May 2003
Revision 4.1

Geode™ SC2200
Thin Client On a Chip

General Description

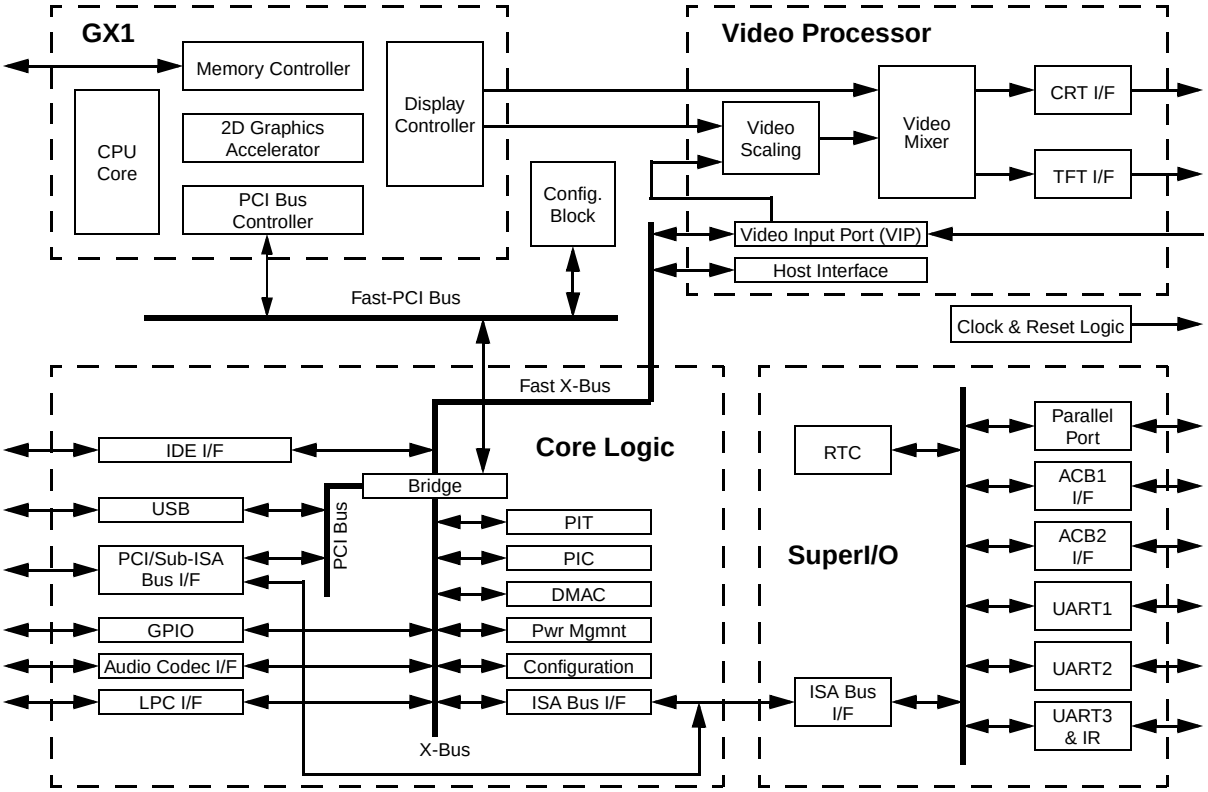
The Geode™ SC2200 Thin Client On a Chip device is a member of the National Semiconductor® IA (Information Appliance) On a Chip family of fully integrated x86 system chips. The Geode SC2200 includes:

- The Geode GX1 processor module combines advanced CPU performance with Intel MMX support, fully accelerated 2D graphics, a 64-bit synchronous DRAM (SDRAM) interface, a PCI bus controller, and a display controller.
- A low-power CRT and TFT Video Processor module with a Video Input Port (VIP), and a hardware video accelerator for scaling, filtering, and color space conversion.
- The Core Logic module includes: PC/AT functionality, a USB interface, an IDE interface, a PCI bus interface, an LPC bus interface, Advanced Configuration Power Interface (ACPI) version 1.0 compliant power management, and an audio codec interface.
- The SuperI/O module has: three Serial Ports (UART1, UART2, and UART3 with fast infrared), a Parallel Port, two ACCESS.bus (ACB) interfaces, and a Real-Time Clock (RTC).

The block diagram shows the relationships between the modules.

These features, combined with the device's small form factor and low power consumption, make it ideal as the core for a thin client application.

Block Diagram



National Semiconductor and Virtual System Architecture are registered trademarks of National Semiconductor Corporation.
Geode and VSA are trademarks of National Semiconductor Corporation.
For a complete listing of National Semiconductor trademarks, please visit www.national.com/trademarks.

© 2003 National Semiconductor Corporation

Features

General Features

- 32-Bit x86 processor, up to 300 MHz, with MMX instruction set support
- Memory controller with 64-bit SDRAM interface
- 2D graphics accelerator
- CRT controller with hardware video accelerator
- CCIR-656 video input port with direct video for full screen display
- PC/AT functionality
- PCI bus controller
- IDE interface, two channels
- USB, three ports, OHCI (OpenHost Controller Interface) version 1.0 compliant
- Audio, AC97/AMC97 version 2.0 compliant
- Virtual System Architecture® (VSA™) technology support
- Power management, ACPI (Advanced Configuration Power Interface) version 1.0 compliant
- Package:
 - 432-Terminal EBGA (Enhanced Ball Grid Array)
 - 481-Terminal TEPBGA (Thermally Enhanced Plastic Ball Grid Array)

GX1 Processor Module

- CPU Core:
 - 32-Bit x86, 300 MHz, with MMX compatible instruction set support
 - 16 KB unified L1 cache
 - Integrated Floating Point Unit (FPU)
 - Re-entrant SMM (System Management Mode) enhanced for VSA
- 2D Graphics Accelerator:
 - Accelerates BitBLTs, line draw and text
 - Supports all 256 raster operations
 - Supports transparent BLTs
 - Runs at core clock frequency
- Memory Controller:
 - 64-Bit SDRAM interface
 - 66 MHz to 100 MHz frequency range
 - Direct interface with CPU/cache, display controller and 2D graphic accelerator
 - Supports clock suspend and power-down/self-refresh
 - Up to two banks of SDRAM (8 devices total) or one SODIMM
- Display Controller:
 - Hardware graphics frame buffer compress/decompress
 - Hardware cursor, 32x32 pixels

Video Processor Module

- Video Accelerator:
 - Flexible video scaling support of up to 8x (horizontally and vertically)
 - Bilinear interpolation filters (with two taps, and eight phases) to smooth output video
- Video/Graphics Mixer:
 - 8-bit value alpha blending
 - Three blending windows with constant alpha value
 - Color key
- Video Input Port (VIP):
 - Video capture or display
 - CCIR-656 format and VESA Video Interface Port Rev 1.1 compliant
 - Lock display timing to video input timing (GenLock)
 - Able to transfer video data into main memory
 - Direct video transfer for full screen display
 - Separate memory location for VBI
- CRT Interface:
 - Uses three 8-bit DACs
 - Supports up to 135 MHz
 - 1280x1024 non-interlaced CRT @ 8 bpp, up to 75 Hz
 - 1024x768 non-interlaced CRT @ 16 bpp, up to 85 Hz
- TFT Interface:
 - Direct connection to TFT panels
 - 800x600 non-interlaced TFT @ 16 bpp graphics, up to 85 Hz
 - 1024x768 non-interlaced TFT @ 16 bpp graphics, up to 75 Hz
 - Connects to Geode CS9211 companion for DSTN panel support
 - TFT on IDE: FPCLK max is 40 MHz
 - TFT on Parallel Port: FPCLK max is 80 MHz

Features (Continued)

Core Logic Module

- Audio Codec Interface:
 - AC97/AMC97 (Rev. 2.0) codec interface
 - Six DMA channels
- PC/AT Functionality:
 - Programmable Interrupt Controller (PIC), 8259A-equivalent
 - Programmable Interval Timer (PIT), 8254-equivalent
 - DMA Controller (DMAC), 8237-equivalent
- Power Management:
 - ACPI 1.0 compliant
 - Sx state control of three power planes
 - Cx/Sx state control of clocks and PLLs
 - Thermal event input
 - Wakeup event support:
 - Three general-purpose events
 - AC97 codec event
 - UART2 RI# signal
 - Infrared (IR) event
- General Purpose I/Os (GPIOs):
 - 27 multiplexed GPIO signals
- Low Pin Count (LPC) Bus Interface:
 - Specification version 1.0 compatible
- PCI Bus Interface:
 - PCI version 2.1 compliant with wakeup capability
 - 32-Bit data path, up to 33 MHz
 - Glueless interface for an external PCI device
 - Fixed priority
 - 3.3V signal support only
- Sub-ISA Bus Interface:
 - Up to 16 MB addressing
 - Supports a chip select for ROM or Flash EPROM boot device
 - Supports either:
 - M-Systems DiskOnChip DOC2000 Flash file system
 - NAND EEPROM
 - Supports up to two chip selects for external I/O devices
 - 8-Bit (optional 16-bit) data bus width
 - Shares balls with PCI signals
 - Is not a subtractive agent
- IDE Interface:
 - Two IDE channels for up to four external IDE devices
 - Supports ATA-33 synchronous DMA mode transfers, up to 33 MB/s
- Universal Serial Bus (USB):
 - USB OpenHCI 1.0 compliant
 - Three ports

SuperI/O Module

- Real-Time Clock (RTC):
 - DS1287, MC146818 and PC87911 compatible
 - Multi-century calendar
- ACCESS.bus (ACB) Interface:
 - Two ACB interface ports
- Parallel Port:
 - EPP 1.9 compliant
 - IEEE 1284 ECP compliant, including level 2
- Serial Port (UART):
 - UART1, 16550A compatible (SIN, SOUT, BOUT pins), used for SmartCard interface
 - UART2, 16550A compatible
 - Enhanced UART with fast Infrared (IR)

Other Features

- High-Resolution Timer:
 - 32-Bit counter with 1 μs count interval
- WATCHDOG Timer:
 - Interfaces to INTR, SMI, Reset
- Clocks:
 - Input (external crystals):
 - 32.768 KHz (internal clock oscillator)
 - 27 MHz (internal clock oscillator)
 - Output:
 - AC97 clock (24.576 MHz)
 - Memory controller clock (66 MHz to 100 MHz)
 - PCI clock (33 MHz)
- JTAG Testability:
 - Bypass, Extest, Sample/Preload, IDcode, Clamp, HIZ
- Voltages:
 - Internal logic: 233/266 MHz @ 1.8V and 300 MHz @ 2.1V
 - Battery: 3V
 - I/O: 3.3V
 - Standby: 3.3V

Table of Contents

1.0 Architecture Overview 10

1.1 GX1 MODULE 11
1.1.1 Memory Controller 11
1.1.2 Fast-PCI Bus 16
1.1.3 Display 16
1.2 VIDEO PROCESSOR MODULE 16
1.2.1 GX1 Module Interface 16
1.2.2 Video Input Port 16
1.2.3 Core Logic Module Interface 16
1.2.4 CRT DAC 16
1.3 CORE LOGIC MODULE 16
1.3.1 Other Interfaces of the Core Logic Module 16
1.4 SUPERI/O MODULE 17
1.5 CLOCK, TIMERS, AND RESET LOGIC 17
1.5.1 Reset Logic 17
1.5.1.1 Power-On Reset 17
1.5.1.2 System Reset 17

2.0 Signal Definitions 18

2.1 BALL ASSIGNMENTS 20
2.2 STRAP OPTIONS 51
2.3 MULTIPLEXING CONFIGURATION 52
2.4 SIGNAL DESCRIPTIONS 56
2.4.1 System Interface 56
2.4.2 Memory Interface Signals 57
2.4.3 Video Port Interface Signals 59
2.4.4 CRT/TFT Interface Signals 60
2.4.5 ACCESS.bus Interface Signals 61
2.4.6 PCI Bus Interface Signals 61
2.4.7 Sub-ISA Interface Signals 66
2.4.8 Low Pin Count (LPC) Bus Interface Signals 67
2.4.9 IDE Interface Signals 68
2.4.10 Universal Serial Bus (USB) Interface Signals 69
2.4.11 Serial Ports (UARTs) Interface Signals 70
2.4.12 Parallel Port Interface Signals 71
2.4.13 Fast Infrared (IR) Port Interface Signals 72
2.4.14 AC97 Audio Interface Signals 73
2.4.15 Power Management Interface Signals 74
2.4.16 GPIO Interface Signals 75
2.4.17 Debug Monitoring Interface Signals 76
2.4.18 JTAG Interface Signals 77
2.4.19 Test and Measurement Interface Signals 78
2.4.20 Power, Ground and No Connections 79

Table of Contents (Continued)

3.0 General Configuration Block 80

3.1 CONFIGURATION BLOCK ADDRESSES 80
3.2 MULTIPLEXING, INTERRUPT SELECTION, AND BASE ADDRESS REGISTERS 81
3.3 WATCHDOG 88
3.3.1 Functional Description 88
3.3.1.1 WATCHDOG Timer 88
3.3.2 WATCHDOG Registers 89
3.3.2.1 Usage Hints 89
3.4 HIGH-RESOLUTION TIMER 90
3.4.1 Functional Description 90
3.4.2 High-Resolution Timer Registers 90
3.4.2.1 Usage Hints 90
3.5 CLOCK GENERATORS AND PLLS 92
3.5.1 27 MHz Crystal Oscillator 93
3.5.2 GX1 Module Core Clock 94
3.5.3 Internal Fast-PCI Clock 94
3.5.4 SuperI/O Clocks 95
3.5.5 Core Logic Module Clocks 95
3.5.6 Video Processor Clocks 95
3.5.7 Clock Registers 95

4.0 SuperI/O Module 98

4.1 FEATURES 99
4.2 MODULE ARCHITECTURE 100
4.3 CONFIGURATION STRUCTURE/ACCESS 101
4.3.1 Index-Data Register Pair 101
4.3.2 Banked Logical Device Registers 101
4.3.3 Default Configuration Setup 102
4.3.4 Address Decoding 102
4.4 STANDARD CONFIGURATION REGISTERS 103
4.4.1 SIO Control and Configuration Registers 106
4.4.2 Logical Device Control and Configuration 107
4.4.2.1 LDN 00h - Real-Time Clock 107
4.4.2.2 LDN 01h - System Wakeup Control 109
4.4.2.3 LDN 02h - Infrared Communication Port or Serial Port 3 110
4.4.2.4 LDN 03h and 08h - Serial Ports 1 and 2 111
4.4.2.5 LDN 05h and 06h - ACCESS.bus Ports 1 and 2 112
4.4.2.6 LDN 07h - Parallel Port 113
4.5 REAL-TIME CLOCK (RTC) 114
4.5.1 Bus Interface 114
4.5.2 RTC Clock Generation 114
4.5.2.1 Internal Oscillator 114
4.5.2.2 External Oscillator 115
4.5.2.3 Timing Generation 115
4.5.2.4 Timekeeping 116
4.5.2.5 Alarms 116
4.5.2.6 Power Supply 117
4.5.2.7 System Power States 118
4.5.2.8 Oscillator Activity 118
4.5.2.9 Interrupt Handling 119
4.5.2.10 Battery-Backed RAMs and Registers 119

Table of Contents (Continued)

4.5.3 RTC Registers 120
4.5.3.1 Usage Hints 124
4.5.4 RTC General-Purpose RAM Map 124
4.6 SYSTEM WAKEUP CONTROL (SWC) 125
4.6.1 Event Detection 125
4.6.1.1 Audio Codec Event 125
4.6.1.2 CEIR Address 125
4.6.2 SWC Registers 126
4.7 ACCESS.BUS INTERFACE 130
4.7.1 Data Transactions 130
4.7.2 Start and Stop Conditions 130
4.7.3 Acknowledge (ACK) Cycle 131
4.7.4 Acknowledge After Every Byte Rule 132
4.7.5 Addressing Transfer Formats 132
4.7.6 Arbitration on the Bus 132
4.7.7 Master Mode 132
4.7.7.1 Master Stop 133
4.7.8 Slave Mode 134
4.7.9 Configuration 134
4.7.10 ACB Registers 135
4.8 LEGACY FUNCTIONAL BLOCKS 138
4.8.1 Parallel Port 138
4.8.1.1 Parallel Port Register and Bit Maps 138
4.8.2 UART Functionality (SP1 and SP2) 140
4.8.2.1 UART Mode Register Bank Overview 140
4.8.2.2 SP1 and SP2 Register and Bit Maps for UART Functionality 140
4.8.3 IR Communications Port (IRCP) / Serial Port 3 (SP3) Functionality 144
4.8.3.1 IR/SP3 Mode Register Bank Overview 144
4.8.3.2 IRCP/SP3 Register and Bit Maps 144

5.0 Core Logic Module 150

5.1 FEATURE LIST 150
5.2 MODULE ARCHITECTURE 151
5.2.1 Fast-PCI Interface to External PCI Bus 152
5.2.1.1 Processor Mastered Cycles 152
5.2.1.2 External PCI Mastered Cycles 152
5.2.1.3 Core Logic Internal or Sub-ISA Mastered Cycles 152
5.2.1.4 External PCI Bus 152
5.2.1.5 Bus Master Request Priority 152
5.2.2 PSERIAL Interface 152
5.2.2.1 Video Retrace Interrupt 152
5.2.3 IDE Controller 153
5.2.3.1 IDE Configuration Registers 153
5.2.3.2 PIO Mode 153
5.2.3.3 Bus Master Mode 154
5.2.3.4 UltraDMA/33 Mode 155
5.2.4 Universal Serial Bus 156
5.2.5 Sub-ISA Bus Interface 156
5.2.5.1 Sub-ISA Bus Cycles 157
5.2.5.2 Sub-ISA Support of Delayed PCI Transactions 157
5.2.5.3 Sub-ISA Bus Data Steering 158
5.2.5.4 I/O Recovery Delays 158
5.2.5.5 ISA DMA 159
5.2.5.6 ROM Interface 160
5.2.5.7 PCI and Sub-ISA Signal Cycle Multiplexing 160

Table of Contents (Continued)

5.2.6 AT Compatibility Logic . . . 161
5.2.6.1 DMA Controller . . . 161
5.2.6.2 Programmable Interval Timer . . . 163
5.2.6.3 Programmable Interrupt Controller . . . 164
5.2.7 I/O Ports 092h and 061h System Control . . . 165
5.2.7.1 I/O Port 092h System Control . . . 165
5.2.7.2 I/O Port 061h System Control . . . 166
5.2.7.3 SMI Generation for NMI . . . 166
5.2.8 Keyboard Support . . . 166
5.2.8.1 Fast Keyboard Gate Address 20 and CPU Reset . . . 166
5.2.9 Power Management Logic . . . 167
5.2.9.1 CPU States . . . 167
5.2.9.2 Sleep States . . . 167
5.2.9.3 Power Planes Control . . . 169
5.2.9.4 Power Management Events . . . 169
5.2.9.5 Usage Hints . . . 170
5.2.10 Power Management Programming . . . 170
5.2.10.1 APM Support . . . 170
5.2.10.2 CPU Power Management . . . 171
5.2.10.3 Peripheral Power Management . . . 172
5.2.10.4 Power Management Programming Summary . . . 175
5.2.11 GPIO Interface . . . 176
5.2.12 Integrated Audio . . . 176
5.2.12.1 Data Transport Hardware . . . 176
5.2.12.2 AC97 Codec Interface . . . 179
5.2.12.3 VSA Technology Support Hardware . . . 180
5.2.12.4 IRQ Configuration Registers . . . 182
5.2.12.5 LPC Interface . . . 182
5.2.12.6 LPC Interface Signal Definitions . . . 183
5.2.12.7 Cycle Types . . . 183
5.2.12.8 LPC Interface Support . . . 183
5.3 REGISTER DESCRIPTIONS . . . 184
5.3.1 PCI Configuration Space and Access Methods . . . 184
5.3.2 Register Summary . . . 185
5.4 CHIPSET REGISTER SPACE . . . 199
5.4.1 Bridge, GPIO, and LPC Registers - Function 0 . . . 199
5.4.1.1 GPIO Support Registers . . . 231
5.4.1.2 LPC Support Registers . . . 235
5.4.2 SMI Status and ACPI Registers - Function 1 . . . 243
5.4.2.1 SMI Status Support Registers . . . 244
5.4.2.2 ACPI Support Registers . . . 253
5.4.3 IDE Controller Registers - Function 2 . . . 263
5.4.3.1 IDE Controller Support Registers . . . 267
5.4.4 Audio Registers - Function 3 . . . 269
5.4.4.1 Audio Support Registers . . . 270
5.4.5 X-Bus Expansion Interface - Function 5 . . . 284
5.4.5.1 X-Bus Expansion Support Registers . . . 287
5.4.6 USB Controller Registers - PCIUSB . . . 289
5.4.7 ISA Legacy Register Space . . . 302

Table of Contents (Continued)

6.0 Video Processor Module 316

6.1 MODULE ARCHITECTURE 317
6.2 FUNCTIONAL DESCRIPTION 318
6.2.1 Video Input Port (VIP) 320
6.2.1.1 Direct Video Mode 320
6.2.1.2 Capture Video Mode 321
6.2.1.3 Capture VBI Mode 323
6.2.2 Video Block 324
6.2.2.1 Video Input Formatter 324
6.2.2.2 Horizontal Downscaler with 4-Tap Filtering 325
6.2.2.3 Line Buffers 326
6.2.2.4 Formatter 326
6.2.2.5 2-Tap Vertical and Horizontal Upscalers 326
6.2.3 Mixer/Blender Block 327
6.2.3.1 YUV to RGB CSC in Video Data Path 328
6.2.3.2 Gamma Correction 328
6.2.3.3 Color/Chroma Key 328
6.2.3.4 Color/Chroma Key and Mixer/Blender 329
6.2.4 VESA DDSC2B and DPMS Support 332
6.2.5 Integrated DACs 332
6.2.6 TFT Interface 333
6.2.7 Integrated PLL 334
6.3 REGISTER DESCRIPTIONS 335
6.3.1 Register Summary 335
6.3.2 Video Processor Registers - Function 4 338
6.3.2.1 Video Processor Support Registers - F4BAR0 340
6.3.2.2 VIP Support Registers - F4BAR2 353

7.0 Debugging and Monitoring 357

7.1 TESTABILITY (JTAG) 357
7.1.1 Mandatory Instruction Support 357
7.1.2 Optional Instruction Support 357
7.1.3 JTAG Chain 357

8.0 Electrical Specifications 358

8.1 GENERAL SPECIFICATIONS 358
8.1.1 Power/Ground Connections and Decoupling 358
8.1.2 Absolute Maximum Ratings 358
8.1.3 Operating Conditions 358
8.1.4 DC Current 360
8.1.4.1 Power State Parameter Definitions 360
8.1.4.2 Definition and Measurement Techniques of SC2200 Current Parameters 360
8.1.4.3 Definition of System Conditions for Measuring On Parameters 361
8.1.4.4 DC Current Measurements 361
8.1.5 Ball Capacitance and Inductance 363
8.1.6 Pull-Up and Pull-Down Resistors 364

Table of Contents (Continued)

8.2 DC CHARACTERISTICS 365
8.2.1 IN_{AB} DC Characteristics 366
8.2.2 IN_{BTN} DC Characteristics 366
8.2.3 IN_{PCI} DC Characteristics 366
8.2.4 IN_{STRP} DC Characteristics 366
8.2.5 IN_{T} DC Characteristics 367
8.2.6 IN_{TS} DC Characteristics 367
8.2.7 IN_{TS1} DC Characteristics 367
8.2.8 IN_{USB} DC Characteristics 367
8.2.9 O_{AC97} DC Characteristics 368
8.2.10 OD_{n} DC Characteristics 368
8.2.11 OD_{PCI} DC Characteristics 368
8.2.12 $O_{p/n}$ DC Characteristics 368
8.2.13 O_{PCI} DC Characteristics 368
8.2.14 O_{USB} DC Characteristics 369
8.2.15 $TS_{p/n}$ DC Characteristics 369
8.2.15.1 Exceptions 369
8.3 AC CHARACTERISTICS 370
8.3.1 Memory Controller Interface 371
8.3.2 Video Port 374
8.3.3 CRT and TFT Interface 375
8.3.4 ACCESS.bus Interface 377
8.3.5 PCI Bus Interface 380
8.3.5.1 Measurement and Test Conditions 383
8.3.6 Sub-ISA Interface 386
8.3.7 LPC Interface 390
8.3.8 IDE Interface 391
8.3.9 Universal Serial Bus (USB) Interface 409
8.3.10 Serial Port (UART) 413
8.3.11 Fast IR Port 414
8.3.12 Parallel Port Interface 415
8.3.12.1 Extended Capabilities Port (ECP) 417
8.3.13 Audio Interface (AC97) 419
8.3.14 Power Management Interface 424
8.3.15 Power-Up Sequencing 425
8.3.16 JTAG Interface 427

9.0 Package Specifications 429

Appendix A Support Documentation 431

A.1 ORDER INFORMATION 431
A.2 DATASHEET REVISION HISTORY 431

1.0 Architecture Overview

As illustrated in Figure 1-1, the SC2200 contains the following modules in one integrated device:

- GX1 Module:
 - — Combines advanced CPU performance with MMX support, fully accelerated 2D graphics, a 64-bit synchronous DRAM (SDRAM) interface and a PCI bus controller. Integrates GX1 silicon revision 8.1.1.
- Video Processor Module:
 - — A low-power CRT and TFT support module with a video input port, and a hardware video accelerator for scaling, filtering and color space conversion.
- Core Logic Module:
 - — Includes PC/AT functionality, an IDE interface, a Universal Serial Bus (USB) interface, ACPI 1.0 compliant power management, and an audio codec interface.
- SuperI/O Module:
 - — Includes two Serial Ports, an Infrared (IR) Port, a Parallel Port, two ACCESS.bus interfaces, and a Real-Time Clock (RTC).

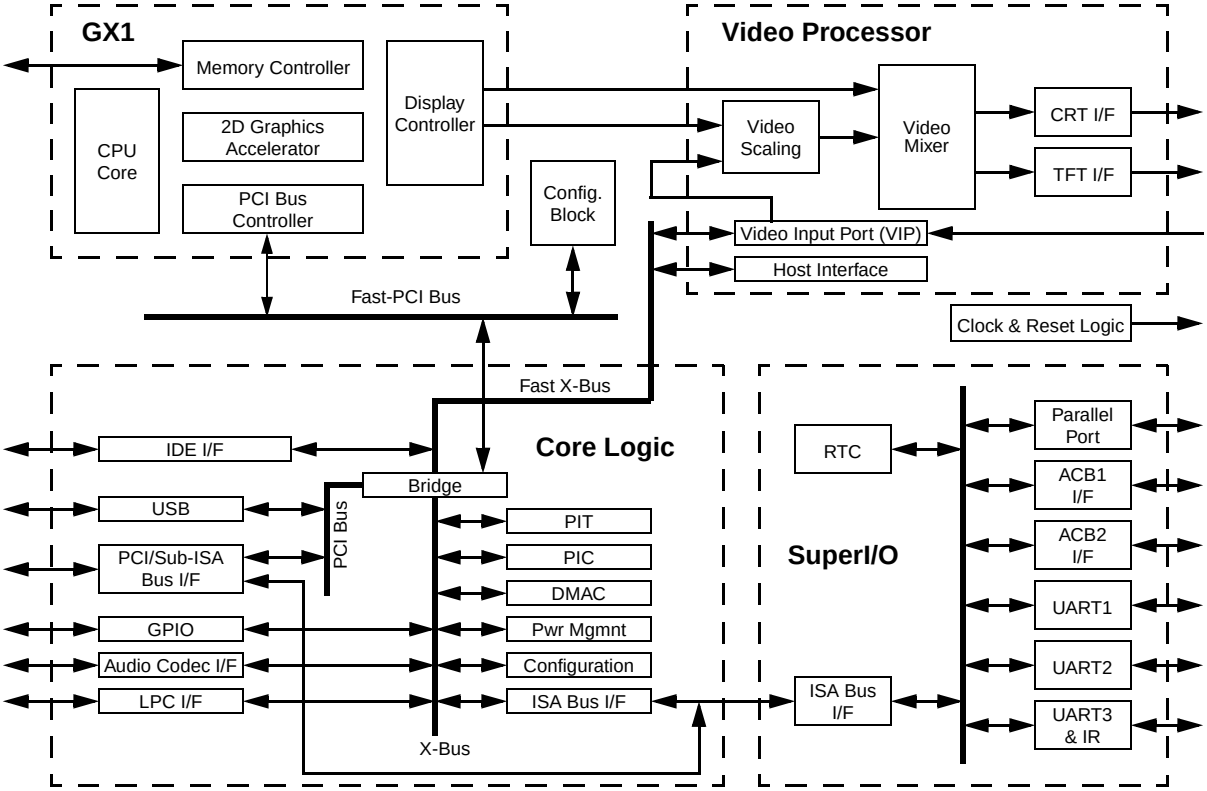


Figure 1-1. SC2200 Block Diagram

1.1 GX1 MODULE

The GX1 processor (silicon revision 8.1.1) is the central module of the SC2200. For detailed information regarding the GX1 module, refer to the G*eode GX1 Processor Series datasheet* and the *Geode GX1 Processor Series Silicon Revision 8.1.1 errata*.

The SC2200's device ID is contained in the GX1 module. Software can detect the revision by reading the DIR0 and DIR1 Configuration registers (see Configuration registers in the G*eode GX1 Processor Series datasheet*). The SC2200 device errata contains the specific values.

1.1.1 Memory Controller

The GX1 module is connected to external SDRAM devices. For more information see Section 2.4.2 "Memory Interface Signals" on page 57, and the "Memory Controller" chapter in the *GX1 Processor Series datasheet*.

There are some differences in the SC2200's memory controller and the stand-alone GX1 processor's memory controller:

1) There is drive strength/slew control in the SC2200 that is not in the GX1. The bits that control this function are in the MC_MEM_CNTRL1 and MC_MEM_CNTRL2 registers. In the GX1 processor, these bits are marked as reserved.

2) The SC2200 supports two banks of memory. The GX1 supports four banks of memory. In addition, the SC2200 supports a maximum of eight devices and the GX1 supports up to 32 devices. With this difference, the MC_BANK_CFG register is different.

Table 1-1 summarizes the 32-bit registers contained in the SC2200's memory controller. Table 1-2 gives detailed register/bit formats.

Table 1-1. SC2200 Memory Controller Register Summary

GX_BASE+ Memory Offset	Width (Bits)	Type	Name/Function	Reset Value
8400h-8403h	32	R/W	**MC_MEM_CNTRL1.** Memory Controller Control Register 1	248C0040h
8404h-8407h	32	R/W	**MC_MEM_CNTRL2.** Memory Controller Control Register 2	00000801h
8408h-840Bh	32	R/W	**MC_BANK_CFG.** Memory Controller Bank Configuration	41104110h
840Ch-840Fh	32	R/W	**MC_SYNC_TIM1.** Memory Controller Synchronous Timing Register 1	2A733225h
8414h-8417h	32	R/W	**MC_GBASE_ADD.** Memory Controller Graphics Base Address Register	00000000h
8418h-841Bh	32	R/W	**MC_DR_ADD.** Memory Controller Dirty RAM Address Register	00000000h
841Ch-841Fh	32	R/W	**MC_DR_ACC.** Memory Controller Dirty RAM Access Register	0000000xh

Architecture Overview (Continued)

Table 1-2. SC2200 Memory Controller Registers

Bit	Description
GX_BASE+ 8400h-8403h	**MC_MEM_CNTRL1 (R/W)** **Reset Value: 248C0040h**
31:30	**MDCTL (MD[63:0] Drive Strength).** 11 is strongest, 00 is weakest.
29	**RSVD (Reserved)** Write as 0.
28:27	**MABACTL (MA[12:0] and BA[1:0] Drive Strength).** 11 is strongest, 00 is weakest.
26	**RSVD (Reserved).** Write as 0.
25:24	**MEMCTL (RASA#, CASA#, WEA#, CS[1:0]#, CKEA, DQM[7:0] Drive Strength).** 11 is strongest, 00 is weakest.
23:22	**RSVD (Reserved).** Write as 0.
21	**RSVD (Reserved).** Must be written as 0. Wait state on the X-Bus x_data during read cycles - for debug only.
20:18	**SDCLKRATE (SDRAM Clock Ratio).** Selects SDRAM clock ratio. 000: Reserved 001: ÷ 2 010: ÷ 2.5 011: ÷ 3 (Default) 100: ÷ 3.5 101: ÷ 4 110: ÷ 4.5 111: ÷ 5 Ratio does not take effect until the SDCLKSTRT bit (bit 17 of this register) transitions from 0 to 1.
17	**SDCLKSTRT (Start SDCLK).** Start operating SDCLK using the new ratio and shift value (selected in bits [20:18] of this register). 0: Clear. 1: Enable. This bit must transition from zero (written to zero) to one (written to one) in order to start SDCLK or to change the shift value.
16:8	**RFSHRATE (Refresh Interval).** This field determines the number of processor core clocks multiplied by 64 between refresh cycles to the DRAM. By default, the refresh interval is 00h. Refresh is turned off by default.
7:6	**RFSHSTAG (Refresh Staggering).** This field determines number of clocks between the RFSH commands to each of the four banks during refresh cycles: 00: 0 SDRAM clocks 01: 1 SDRAM clocks (Default) 10: 2 SDRAM clocks 11: 4 SDRAM clocks Staggering is used to help reduce power spikes during refresh by refreshing one bank at a time. If only one bank is installed, this field must be written as 00.
5	**2CLKADDR (Two Clock Address Setup).** Assert memory address for one extra clock before CS# is asserted. 0: Disable. 1: Enable. This can be used to compensate for address setup at high frequencies and/or high loads.
4	**RFSHTST (Test Refresh).** This bit, when set high, generates a refresh request. This bit is only used for testing purposes.
3	**XBUSARB (X-Bus Round Robin).** When enabled, processor, graphics pipeline and non-critical display controller requests are arbitrated at the same priority level. When disabled, processor requests are arbitrated at a higher priority level. High priority display controller requests always have the highest arbitration priority. 0: Enable. 1: Disable.
2	**SMM_MAP (SMM Region Mapping).** Maps the SMM memory region at GX_BASE+400000 to physical address A0000 to BFFFF in SDRAM. 0: Disable. 1: Enable.
1	**RSVD (Reserved).** Write as 0.
0	**SDRAMPRG (Program SDRAM).** When this bit is set, the memory controller will program the SDRAM MRS register using LTMODE in MC_SYNC_TIM1. This bit must transition from zero (written to zero) to one (written to one) in order to program the SDRAM devices.

Table 1-2. SC2200 Memory Controller Registers (Continued)

Bit	Description
GX_BASE+8404h-8407h	**MC_MEM_CNTRL2 (R/W)** — **Reset Value: 00000801h**
31:14	**RSVD (Reserved).** Write as 0.
13:12	**SDCLKCTL (SDCLK High Drive/Slew Control).** Controls the high drive and slew rate of SDCLK[3:0] and SDCLK_OUT. 11 is strongest, 00 is weakest.
11	**RSVD (Reserved).** Write as 0.
10	**SDCLKOMSK# (Enable SDCLK_OUT).** Turns on the output. 0: Enable. 1: Disable.
9	**SDCLK3MSK# (Enable SDCLK3).** Turns on the output. 0: Enable. 1: Disable.
8	**SDCLK2MSK# (Enable SDCLK2).** Turns on the output. 0: Enable. 1: Disable.
7	**SDCLK1MSK# (Enable SDCLK1).** Turns on the output. 0: Enable. 1: Disable.
6	**SDCLK0MSK# (Enable SDCLK0).** Turns on the output. 0: Enable. 1: Disable.
5:3	**SHFTSDCLK (Shift SDCLK).** This function allows shifting SDCLK to meet SDRAM setup and hold time requirements. The shift function will not take effect until the SDCLKSTRT bit (bit 17 of MC_MEM_CNTRL1) transitions from 0 to 1: 000: No shift — 100: Shift 2 core clocks 001: Shift 0.5 core clock — 101: Shift 2.5 core clocks 010: Shift 1 core clock — 110: Shift 3 core clocks 011: Shift 1.5 core clock — 111: Reserved
2	**RSVD (Reserved).** Write as 0.
1	**RD (Read Data Phase).** Selects if read data is latched one or two core clock after the rising edge of SDCLK. 0: 1 Core clock. 1: 2 Core clocks.
0	**FSTRDMSK (Fast Read Mask).** Do not allow core reads to bypass the request FIFO. 0: Disable. 1: Enable.
GX_BASE+8408h-840Bh	**MC_BANK_CFG (R/W)** — **Reset Value: 41104110h**
31:16	**RSVD (Reserved).** Write as 0070h
15	**RSVD (Reserved).** Write as 0.
14	**SODIMM_MOD_BNK (SODIMM Module Banks - Banks 0 and 1).** Selects number of module banks installed per SODIMM for SODIMM: 0: 1 Module bank (Bank 0 only). 1: 2 Module banks (Bank 0 and 1).
13	**RSVD (Reserved).** Write as 0.
12	**SODIMM_COMP_BNK (SODIMM Component Banks - Banks 0 and 1).** Selects the number of component banks per module bank for SODIMM: 0: 2 Component banks. 1: 4 Component banks. Banks 0 and 1 must have the same number of component banks.
11	**RSVD (Reserved).** Write as 0.
10:8	**SODIMM_SZ (SODIMM Size - Banks 0 and 1).** Selects the size of SODIMM: 000: 4 MB — 010: 16 MB — 100: 64 MB — 110: 256 MB 001: 8 MB — 011: 32 MB — 101: 128 MB — 111: 512 MB This size is the total of both banks 0 and 1. Also, banks 0 and 1 must be the same size.
7	**RSVD (Reserved).** Write as 0.

Architecture Overview (Continued)

Table 1-2. SC2200 Memory Controller Registers (Continued)

Bit	Description
6:4	**SODIMM_PG_SZ (SODIMM Page Size - Banks 0 and 1).** Selects the page size of SODIMM: 000: 1 KB 010: 4 KB 1xx: 16 KB 001: 2 KB 011: 8 KB 111: SODIMM not installed Both banks 0 and 1 must have the same page size.
3:0	**RSVD (Reserved).** Write as 0.
GX_BASE+840Ch-840Fh	**MC_SYNC_TIM1 (R/W)** **Reset Value: 2A733225h**
31	**RSVD (Reserved).** Write as 0.
30:28	**LTMODE (CAS Latency).** CAS latency is the delay, in SDRAM clock cycles, between the registration of a read command and the availability of the first piece of output data. This parameter significantly affects system performance. Optimal setting should be used. If an SODIMM is used, BIOS can interrogate EEPROM across the ACCESS.bus interface to determine this value: 000: Reserved 010: 2 CLK 100: 4 CLK 110: 6 CLK 001: Reserved 011: 3 CLK 101: 5 CLK 111: 7 CLK This field will not take effect until SDRAMPRG (bit 0 of MC_MEM_CNTRL1) transitions from 0 to 1.
27:24	**RC (RFSH to RFSH/ACT Command Period, tRC).** Minimum number of SDRAM clock between RFSH and RFSH/ACT commands: 0000: Reserved 0100: 5 CLK 1000: 9 CLK 1100: 13 CLK 0001: 2 CLK 0101: 6 CLK 1001: 10 CLK 1101: 14 CLK 0010: 3 CLK 0110: 7 CLK 1010: 11 CLK 1110: 15 CLK 0011: 4 CLK 0111: 8 CLK 1011: 12 CLK 1111: 16 CLK
23:20	**RAS (ACT to PRE Command Period, tRAS).** Minimum number of SDRAM clocks between ACT and PRE commands: 0000: Reserved 0100: 5 CLK 1000: 9 CLK 1100: 13 CLK 0001: 2 CLK 0101: 6 CLK 1001: 10 CLK 1101: 14 CLK 0010: 3 CLK 0110: 7 CLK 1010: 11 CLK 1110: 15 CLK 0011: 4 CLK 0111: 8 CLK 1011: 12 CLK 1111: 16 CLK
19	**RSVD (Reserved).** Write as 0.
18:16	**RP (PRE to ACT Command Period, tRP).** Minimum number of SDRAM clocks between PRE and ACT commands: 000: Reserved 010: 2 CLK 100: 4 CLK 110: 6 CLK 001: 1 CLK 011: 3 CLK 101: 5 CLK 111: 7 CLK
15	**RSVD (Reserved).** Write as 0.
14:12	**RCD (Delay Time ACT to READ/WRT Command, tRCD).** Minimum number of SDRAM clock between ACT and READ/WRT commands. This parameter significantly affects system performance. Optimal setting should be used: 000: Reserved 010: 2 CLK 100: 4 CLK 110: 6 CLK 001: 1 CLK 011: 3 CLK 101: 5 CLK 111: 7 CLK
11	**RSVD (Reserved).** Write as 0.
10:8	**RRD (ACT(0) to ACT(1) Command Period, tRRD).** Minimum number of SDRAM clocks between ACT and ACT command to two different component banks within the same module bank. The memory controller does not perform back-to-back Activate commands to two different component banks without a READ or WRITE command between them. Hence, this field should be written as 001.
7	**RSVD (Reserved).** Write as 0.
6:4	**DPL (Data-in to PRE command period, tDPL).** Minimum number of SDRAM clocks from the time the last write datum is sampled till the bank is precharged: 000: Reserved 010: 2 CLK 100: 4 CLK 110: 6 CLK 001: 1 CLK 011: 3 CLK 101: 5 CLK 111: 7 CLK
3:0	**RSVD (Reserved).** Leave unchanged. Always returns a 101h.

Note: Refer to the SDRAM manufacturer's specification for more information on component banks.

Table 1-2. SC2200 Memory Controller Registers (Continued)

Bit	Description
GX_BASE+8414h-8417h	**MC_GBASE_ADD (R/W)** **Reset Value: 00000000h**
31:18	**RSVD (Reserved).** Write as 0.
17	**TE (Test Enable TEST[3:0]).** 0: TEST[3:0] are driven low (normal operation). 1: TEST[3:0] pins are used to output test information.
16	**TECTL (Test Enable Shared Control Pins).** 0: RASB#, CASB#, CKEB, WEB# (normal operation). 1: RASB#, CASB#, CKEB, WEB# are used to output test information.
15:12	**SEL (Select).** This field is used for debug purposes only and should be left at zero for normal operation.
11	**RSVD (Reserved).** Write as 0.
10:0	**GBADD (Graphics Base Address).** This field indicates the graphics memory base address, which is programmable on 512 KB boundaries. This field corresponds to address bits [29:19]. Note that BC_DRAM_TOP must be set to a value lower than the Graphics Base Address.
GX_BASE+8418h-841Bh	**MC_DR_ADD (R/W)** **Reset Value: 00000000h**
31:10	**RSVD (Reserved).** Write as 0.
9:0	**DRADD (Dirty RAM Address).** This field is the address index that is used to access the Dirty RAM with the MC_DR_ACC register. This field does not auto increment.
GX_BASE+841Ch-841Fh	**MC_DR_ACC (R/W)** **Reset Value: 0000000xh**
31:2	**RSVD (Reserved).** Write as 0.
1	**D (Dirty Bit).** This bit is read/write accessible.
0	**V (Valid Bit).** This bit is read/write accessible.

Architecture Overview (Continued)

1.1.2 Fast-PCI Bus

The GX1 module communicates with the Core Logic module via a Fast-PCI bus that can work at up to 66 MHz. The Fast-PCI bus is internal for the SC2200 and is connected to the General Configuration Block.

This bus supports seven bus masters. The requests (REQs) are fixed in priority. The seven bus masters in order of priority are:

1) VIP
2) IDE Channel 0
3) IDE Channel 1
4) Audio
5) USB
6) External REQ0#
7) External REQ1#

1.1.3 Display

The GX1 module generates display timing, and controls internal signals CRT_VSYNC and CRT_HSYNC of the Video Processor module.

The GX1 module interfaces with the Video Processor via a video data bus and a graphics data bus.

- **Video data.** The GX1 module uses the core clock, divided by 2 or 4 (typically 100 - 133 MHz). It drives the video data using this clock. Internal signals VID_VAL and VID_RDY are used as data-flow handshake signals between the GX1 module and the Video Processor.
- **Graphics data.** The GX1 module uses internal signal DCLK, supplied by the PLL of the Video Processor, to drive the 18-bit graphics-data bus of the Video Processor. Each six bits of this bus define a different color. Each of these six-bit color definitions is expanded (by adding two zero LSB lines) to form an eight-bit bus, at the Video Processor.

For more information about the GX1 module's interface to the Video Processor, see the "Display Controller" chapter in the *GX1 Processor Series datasheet*.

1.2 VIDEO PROCESSOR MODULE

The Video Processor provides high resolution and graphics for a CRT or TFT/DSTN interface. The following paragraphs provide a summary of how this Video Processor interfaces with the other modules of the SC2200. For detailed information about the Video Processor, see Section 6.0 "Video Processor Module" on page 316.

1.2.1 GX1 Module Interface

The Video Processor is connected to the GX1 module in the following way:

- The Video Processor DOTCLK output signal is used as the GX1 module's DCLK input signal.
- The GX1 module's PCLK output signal is used as the GFXCLK input signal of the Video Processor.

1.2.2 Video Input Port

The Video Input Port (VIP) within the Video Processor contains a standard interface that is typically connected to a media processor or TV encoder. The clock is supplied by the externally connected device; typically at 27 MHz.

Video input can be sent to the GX1 module's video frame buffer (Capture Video mode) or can be used directly (Direct Video mode).

1.2.3 Core Logic Module Interface

The Video Processor interfaces to the Core Logic module for accessing PCI function configuration registers.

1.2.4 CRT DAC

The Video Processor drives three CRT DACs with up to 135M pixels per second.

The interface for these DACs can be monitored via external balls of the SC2200. For more information, see Section 2.4.4 "CRT/TFT Interface Signals" on page 60.

1.3 CORE LOGIC MODULE

The Core Logic module is described in detail in Section 5.0 "Core Logic Module" on page 150.

The Core Logic module is connected to the Fast-PCI bus. It uses signal AD28 as the IDSEL for all PCI configuration functions except for USB which uses AD29.

1.3.1 Other Interfaces of the Core Logic Module

All the following interfaces of the Core Logic module are implemented via external balls of the SC2200. Each interface is listed below with a reference to the descriptions of the relevant balls.

- IDE: See Section 2.4.9 "IDE Interface Signals" on page 68.
- AC97: See Section 2.4.14 "AC97 Audio Interface Signals" on page 73.
- PCI: See Section 2.4.6 "PCI Bus Interface Signals" on page 61.
- USB: See Section 2.4.10 "Universal Serial Bus (USB) Interface Signals" on page 69. The USB function uses signal AD29 as the IDSEL for PCI configuration.

- LPC: See Section 2.4.8 "Low Pin Count (LPC) Bus Interface Signals" on page 67.
- Sub-ISA: See Section 2.4.7 "Sub-ISA Interface Signals" on page 66, Section 5.2.5 "Sub-ISA Bus Interface" on page 156, and Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81
- GPIO: See Section 2.4.16 "GPIO Interface Signals" on page 75.
- More detailed information about each of these interfaces is provided in Section 5.2 "Module Architecture" on page 151.
- Super/IO Block Interfaces: See Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81, Section 2.4.5 "ACCESS.bus Interface Signals" on page 61, Section 2.4.13 "Fast Infrared (IR) Port Interface Signals" on page 72, and Section 2.4.12 "Parallel Port Interface Signals" on page 71.

The Core Logic module interface to the GX1 module consists of seven miscellaneous connections, the PCI bus interface signals, plus the display controller connections. Note that the PC/AT legacy signals NMI, WM_RST, and A20M are all virtual functions executed in SMM (System Management Mode) by the BIOS.

- PSERIAL is a one-way serial bus from the GX1 to the Core Logic module used to communicate power-management states and VSYNC information for VGA emulation.
- IRQ13 is an input from the processor indicating that a floating point error was detected and that INTR should be asserted.
- INTR is the level output from the integrated 8259A PICs and is asserted if an unmasked interrupt request (IRQn) is sampled active.
- SMI# is a level-sensitive interrupt to the GX1 that can be configured to assert on a number of different system events. After an SMI# assertion, SMM is entered and program execution begins at the base of the SMM address space. Once asserted, SMI# remains active until the SMI source is cleared.
- SUSP# and SUSPA# are handshake signals for implementing CPU Clock Stop and clock throttling.
- CPU_RST resets the CPU and is asserted for approximately 100 µs after the negation of POR#.
- PCI bus interface signals.

1.4 SUPERI/O MODULE

The SuperI/O (SIO) module is a member of National Semiconductor's SuperI/O family of integrated PC peripherals. It is a PC98 and ACPI compliant SIO that offers a single-cell solution to the most commonly used ISA peripherals.

The SIO module incorporates: two Serial Ports, an Infrared Communication Port that supports FIR, MIR, HP-SIR, Sharp-IR, and Consumer Electronics-IR, a full IEEE 1284 Parallel Port, two ACCESS.bus Interface (ACB) ports, System Wakeup Control (SWC), and a Real-Time Clock (RTC) that provides RTC timekeeping.

1.5 CLOCK, TIMERS, AND RESET LOGIC

In addition to the four main modules (i.e., GX1, Core Logic, Video Processor and SIO) that make up the SC2200, the following blocks of logic have also been integrated into the SC2200:

- Clock Generators as described in Section 3.5 "Clock Generators and PLLs" on page 92.
- Configuration Registers as described in Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81.
- A WATCHDOG timer as described in Section 3.3 "WATCHDOG" on page 88.
- A High-Resolution timer as described in Section 3.4 "High-Resolution Timer" on page 90.

1.5.1 Reset Logic

This section provides a description of the reset flow of the SC2200.

1.5.1.1 Power-On Reset

Power-On reset is triggered by assertion of the POR# signal. Upon power-on reset, the following things happen:

- Strap balls are sampled.
- PLL4, PLL5, and PLL6 are reset, disabling their output. When the POR# signal is negated, the clocks lock and then each PLL outputs its clock. PLL6 is the last clock generator to output a clock. See Section 3.5 "Clock Generators and PLLs" on page 92.
- Certain WATCHDOG and High-Resolution Timer register bits are cleared.

1.5.1.2 System Reset

System reset causes signal PCIRST# to be issued, thus triggering reset of all PCI and LPC agents. A system reset is triggered by any of the following events:

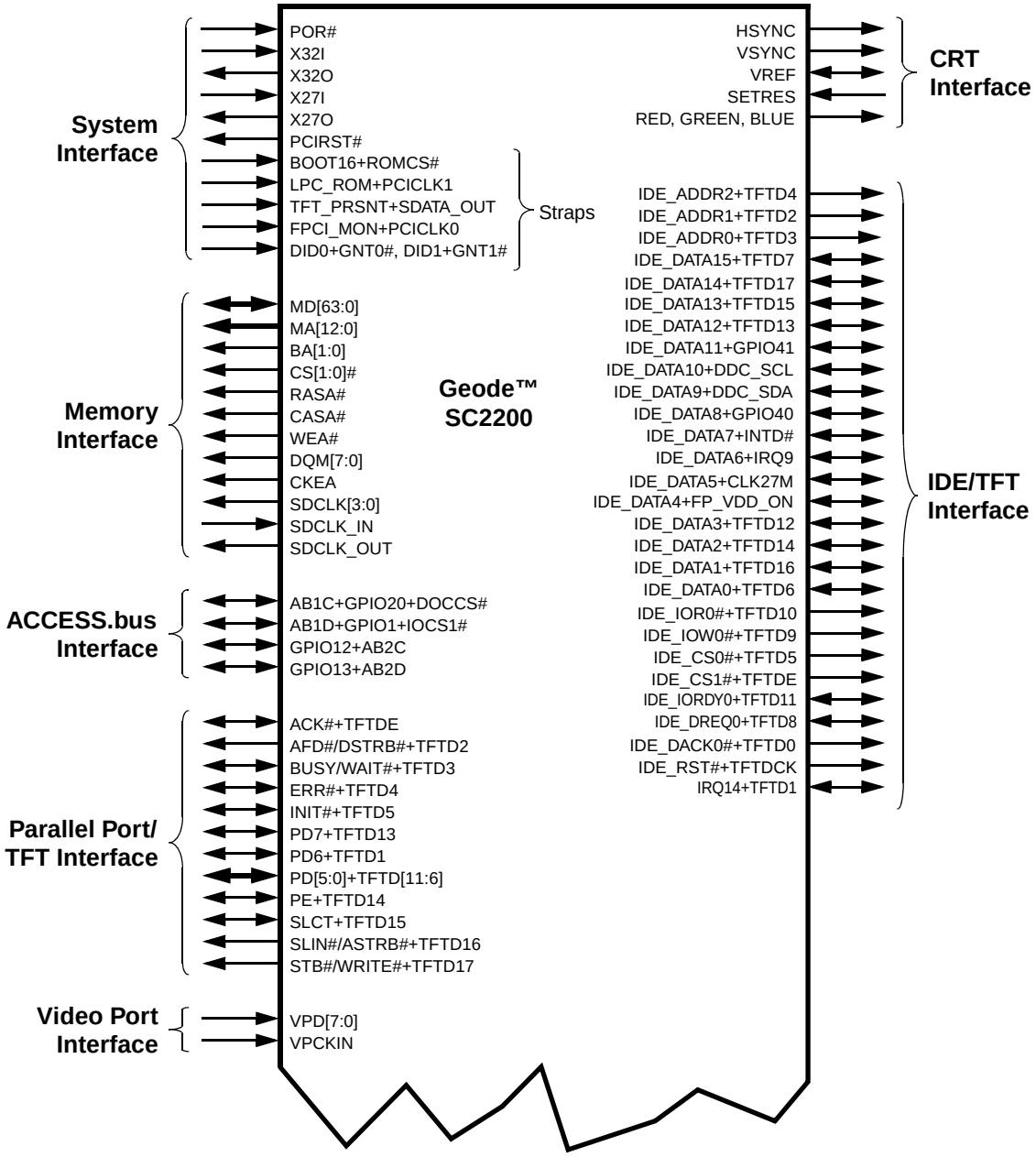
- Power-on, as indicated by POR# signal assertion.
- A WATCHDOG reset event (see Section 3.3.2 "WATCHDOG Registers" on page 89).
- Software initiated system reset.

2.0 Signal Definitions

This section defines the signals and describes the external interface of the SC2200. Figure 2-1 shows the signals organized by their functional groups. Where signals are multiplexed, the default signal name is listed first and is separated by a plus sign (+). A slash (/) in a signal name means that the function is always enabled and available (i.e., cycle multiplexed).

The remaining subsections of this chapter describe:

- Section 2.1 "Ball Assignments": Provides a ball assignment diagram and tables listing the signals sorted according to ball number and alphabetically by signal name.
- Section 2.2 "Strap Options": Several balls are read at power-up that set up the state of the SC2200. This section provides details regarding those balls.
- Section 2.3 "Multiplexing Configuration": Lists multiplexing options and their configurations.
- Section 2.4 "Signal Descriptions": Detailed descriptions of each signal according to functional group.



Note: Straps are not the default signal, shown with system signals for reader convenience. However, they are also listed with the appropriate functional group.

Figure 2-1. Signal Groups

Signal Definitions (Continued)

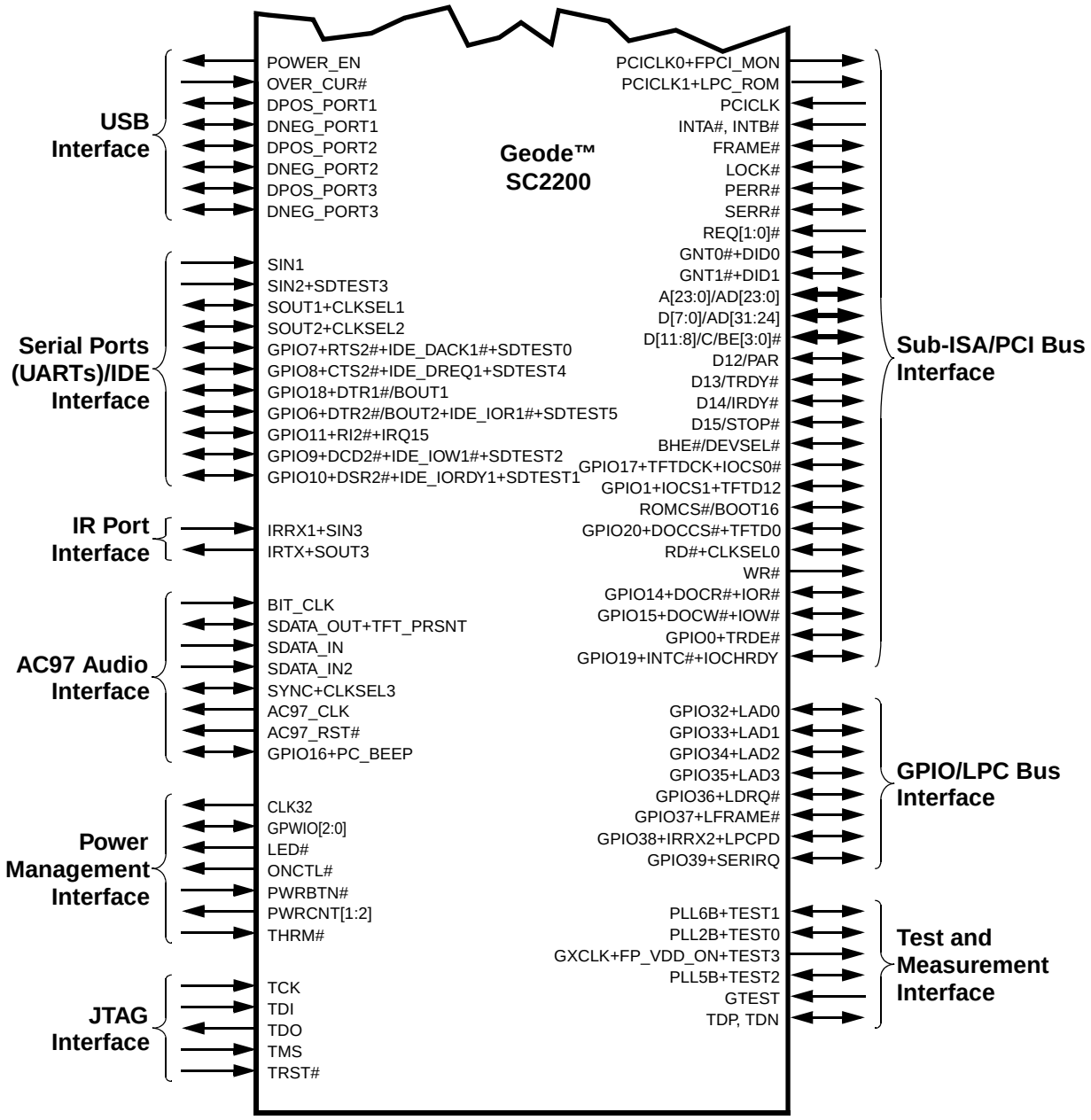


Figure 2-1. Signal Groups (Continued)

Signal Definitions (Continued)

2.1 BALL ASSIGNMENTS

The SC2200 is highly configurable as illustrated in Figure 2-1 on page 18. Strap options and register programming are used to set various modes of operation and specific signals on specific balls. This section describes which signals are available on which balls and provides configuration information:

- Figure 2-2 on page 21 and Figure 2-3 on page 36: Illustrations of EBGA and TEPBGA ball assignments.
- Table 2-2 on page 22 and Table 2-4 on page 37: Lists signals according to ball number. Power Rail, Signal Type, Buffer Type and, where relevant, Pull-Up or Pull-Down resistors are indicated for each ball in this table. For multiplexed balls, the necessary configuration for each signal is listed as well.
- Table 2-3 on page 32 and Table 2-5 on page 47: Quick reference signal list sorted alphabetically - listing all signal names and ball numbers.The tables in this chapter use several common abbreviations. Table 2-1 lists the mnemonics and their meanings

Notes:

1) For each GPIO signal, there is an optional pull-up resistor on the relevant ball. After system reset, the pull-up is present.

 This pull-up resistor can be disabled via registers in the Core Logic module. The configuration is without regard to the selected ball function (except for GPIO12, GPIO13, and GPIO16). Alternate functions for GPIO12, GPIO13, and GPIO16 control pull-up resistors.

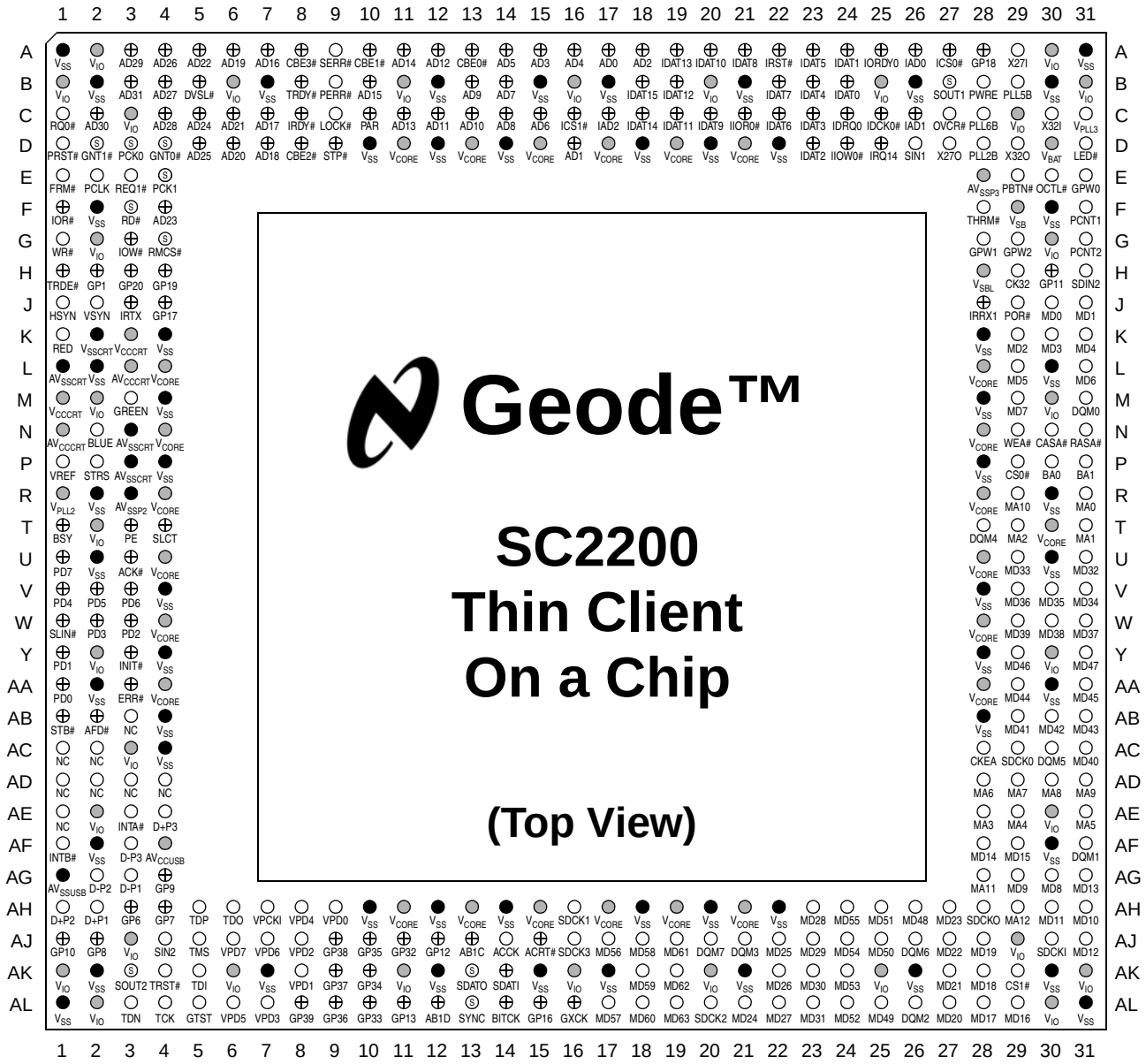
 For more information, see Section 5.4.1 "Bridge, GPIO, and LPC Registers - Function 0" on page 199.

2) Configuration settings listed in this table are with regard to the Pin Multiplexing Register (PMR). See Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81 for a detailed description of this register.

Table 2-1. Signal Definitions Legend

Mnemonic	Definition
A	Analog
AV_{SS}	Ground ball: Analog
AV_{CC}	Power ball: Analog
GCB	General Configuration Block registers. Refer to Section 3.0 "General Configuration Block" on page 80. Location of the General Configuration Block cannot be determined by software. See *SC2200 Thin Client On a Chip device errata.*
I	Input ball
I/O	Bidirectional ball
MCR[x]	Miscellaneous Configuration Register Bit x: A register, located in the GCB. Refer to Section 3.1 "Configuration Block Addresses" on page 80 for further details.
O	Output ball
OD	Open-drain
PD	Pull-down
PMR[x]	Pin Multiplexing Register Bit x: A register, located in the GCB, used to configure balls with multiple functions. Refer to Section 3.1 "Configuration Block Addresses" on page 80 for further details.
PU	Pull-up
TS	TRI-STATE
V_{CORE}	Power ball: 1.2V
V_{IO}	Power ball: 3.3V
V_{SS}	Ground ball
#	The # symbol in a signal name indicates that the active or asserted state occurs when the signal is at a low voltage level. Otherwise, the signal is asserted when at a high voltage level.
/	A / in a signal name indicates both functions are always enabled (i.e., cycle multiplexed).
+	A + in signal name indicates the function is available on the ball, but that either strapping options or register programming is required to select the desired function.

Signal Definitions (Continued)



Note: Signal names have been abbreviated in this figure due to space constraints.

- ● = GND Ball
- ○ = PWR Ball
- Ⓢ = Strap Option Ball
- ⊕ = Multiplexed Ball

Figure 2-2. 432-EBGA Ball Assignment Diagram

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
A1	V_{SS}	GND	---	---	---
A2	V_{IO}	PWR	---	---	---
A3	AD29	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D5	I/O	IN_{PCI}, O_{PCI}		
A4	AD26	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D2	I/O	IN_{PCI}, O_{PCI}		
A5	AD22	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A22	O	O_{PCI}		
A6	AD19	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A19	O	O_{PCI}		
A7	AD16	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A16	O	O_{PCI}		
A8	C/BE3#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D11	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
A9	SERR#	I/O ($PU_{22.5}$)	IN_{PCI}, OD_{PCI}	V_{IO}	---
A10	C/BE1#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D9	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
A11	AD14	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A14	O	O_{PCI}		
A12	AD12	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A12	O	O_{PCI}		
A13	C/BE0#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D8	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
A14	AD5	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A5	O	O_{PCI}		
A15	AD3	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A3	O	O_{PCI}		
A16	AD4	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A4	O	O_{PCI}		
A17	AD0	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A0	O	O_{PCI}		
A18	AD2	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A2	O	O_{PCI}		
A19	IDE_DATA13	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD15	O	$O_{1/4}$		PMR[24] = 1
A20[2]	IDE_DATA10	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	DDC_SCL	O	OD_4		PMR[24] = 1
A21	IDE_DATA8	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	GPIO40	I/O	IN_{TS1}, $O_{1/4}$		PMR[24] = 1
A22	IDE_RST#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTDCK	O	$O_{1/4}$		PMR[24] = 1
A23	IDE_DATA5	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	CLK27M	O	$O_{1/4}$		PMR[24] = 1
A24	IDE_DATA1	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD16	O	$O_{1/4}$		PMR[24] = 1
A25	IDE_IORDY0	I	IN_{TS1}	V_{IO}	PMR[24] = 0
	TFTD11	O	$O_{1/4}$		PMR[24] = 1
A26	IDE_ADDR0	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD3	O	$O_{1/4}$		PMR[24] = 1
A27	IDE_CS0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD5	O	$O_{1/4}$		PMR[24] = 1
A28	GPIO18	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[16] = 0
	DTR1#/BOUT1	O ($PU_{22.5}$)	$O_{8/8}$		PMR[16] =1
A29	X27I	I	WIRE	V_{IO}	---
A30	V_{IO}	PWR	---	---	---
A31	V_{SS}	GND	---	---	---
B1	V_{IO}	PWR	---	---	---
B2	V_{SS}	GND	---	---	---
B3	AD31	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D7	I/O	IN_{PCI}, O_{PCI}		
B4	AD27	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D3	I/O	IN_{PCI}, O_{PCI}		
B5	DEVSEL#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	BHE#	O	O_{PCI}		
B6	V_{IO}	PWR	---	---	---
B7	V_{SS}	GND	---	---	---
B8	TRDY#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D13	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
B9	PERR#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
B10	AD15	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A15	O	O_{PCI}		
B11	V_{IO}	PWR	---	---	---
B12	V_{SS}	GND	---	---	---
B13	AD9	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A9	O	O_{PCI}		
B14	AD7	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A7	O	O_{PCI}		
B15	V_{SS}	GND	---	---	---
B16	V_{IO}	PWR	---	---	---
B17	V_{SS}	GND	---	---	---
B18	IDE_DATA15	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD7	O	$O_{1/4}$		PMR[24] = 1
B19	IDE_DATA12	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD13	O	$O_{1/4}$		PMR[24] = 1
B20	V_{IO}	PWR	---	---	---
B21	V_{SS}	GND	---	---	---
B22	IDE_DATA7	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	INTD#	I	IN_{TS}		PMR[24] = 1
B23	IDE_DATA4	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	FP_VDD_ON	O	$O_{1/4}$		PMR[24] = 1
B24	IDE_DATA0	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD6	O	$O_{1/4}$		PMR[24] = 1
B25	V_{IO}	PWR	---	---	---
B26	V_{SS}	GND	---	---	---
B27	SOUT1	O	$O_{8/8}$	V_{IO}	---
	CLKSEL1	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
B28	POWER_EN	O	$O_{1/4}$	V_{IO}	---
B29	PLL5B	I/O	IN_T, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST2	O	$O_{2/5}$		PMR[29] = 1
B30	V_{SS}	GND	---	---	---
B31	V_{IO}	PWR	---	---	---
C1	REQ0#	I ($PU_{22.5}$)	INPCI	V_{IO}	---
C2	AD30	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D6	I/O	IN_{PCI}, O_{PCI}		
C3	V_{IO}	PWR	---	---	---
C4	AD28	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D4	I/O	IN_{PCI}, O_{PCI}		
C5	AD24	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D0	I/O	IN_{PCI}, O_{PCI}		
C6	AD21	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A21	O	O_{PCI}		
C7	AD17	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A17	O	O_{PCI}		
C8	IRDY#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D14	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
C9	LOCK#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
C10	PAR	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D12	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
C11	AD13	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A13	O	O_{PCI}		
C12	AD11	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A11	O	O_{PCI}		
C13	AD10	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A10	O	O_{PCI}		
C14	AD8	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A8	O	O_{PCI}		
C15	AD6	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A6	O	O_{PCI}		
C16	IDE_CS1#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTDE	O	$O_{1/4}$		PMR[24] = 1
C17	IDE_ADDR2	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD4	O	$O_{1/4}$		PMR[24] = 1
C18	IDE_DATA14	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD17	O	$O_{1/4}$		PMR[24] = 1
C19	IDE_DATA11	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	GPIO41	I/O	IN_{TS1}, $O_{1/4}$		PMR[24] = 1

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
C20[2]	IDE_DATA9	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	DDC_SDA	I/O	IN_T, OD_4		PMR[24] = 1
C21	IDE_IOR0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD10	O	$O_{1/4}$		PMR[24] = 1
C22	IDE_DATA6	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	IRQ9	I	IN_{TS1}		PMR[24] = 1
C23	IDE_DATA3	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD12	O	$O_{1/4}$		PMR[24] = 1
C24	IDE_DREQ0	I	IN_{TS1}	V_{IO}	PMR[24] = 0
	TFTD8	O	$O_{1/4}$		PMR[24] = 1
C25	IDE_DACK0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD0	O	$O_{1/4}$		PMR[24] = 1
C26	IDE_ADDR1	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD2	O	$O_{1/4}$		PMR[24] = 1
C27	OVER_CUR#	I	IN_{TS}	V_{IO}	---
C28	PLL6B	I/O	IN_{TS}, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST1	O	$O_{2/5}$		PMR[29] = 1
C29	V_{IO}	PWR	---	---	---
C30	X32I	I	WIRE	V_{BAT}	---
C31	V_{PLL3}	PWR	---	---	---
D1	PCIRST#	O	O_{PCI}	V_{IO}	---
D2	GNT1#	O	O_{PCI}	V_{IO}	---
	DID1	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
D3	PCICLK0	O	O_{PCI}	V_{IO}	---
	FPCI_MON	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
D4	GNT0#	O	O_{PCI}	V_{IO}	---
	DID0	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
D5	AD25	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D1	I/O	IN_{PCI}, O_{PCI}		
D6	AD20	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A20	O	O_{PCI}		
D7	AD18	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A18	O	O_{PCI}		
D8	C/BE2#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D10	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
D9	STOP#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D15	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
D10	V_{SS}	GND	---	---	---
D11	V_{CORE}	PWR	---	---	---
D12	V_{SS}	GND	---	---	---
D13	V_{CORE}	PWR	---	---	---
D14	V_{SS}	GND	---	---	---
D15	V_{CORE}	PWR	---	---	---
D16	AD1	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A1	O	O_{PCI}		
D17	V_{CORE}	PWR	---	---	---
D18	V_{SS}	GND	---	---	---
D19	V_{CORE}	PWR	---	---	---
D20	V_{SS}	GND	---	---	---
D21	V_{CORE}	PWR	---	---	---
D22	V_{SS}	GND	---	---	---
D23	IDE_DATA2	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD14	O	$O_{1/4}$		PMR[24] = 1
D24	IDE_IOW0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD9	O	$O_{1/4}$		PMR[24] = 1
D25	IRQ14	I	IN_{TS1}	V_{IO}	PMR[24] = 0
	TFTD1	O	$O_{1/4}$		PMR[24] = 1
D26	SIN1	I	IN_{TS}	V_{IO}	---
D27	X27O	O	WIRE	V_{IO}	---
D28	PLL2B	I/O	IN_T, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST0	O	$O_{2/5}$		PMR[29] = 1
D29	X32O	O	WIRE	V_{BAT}	---
D30	V_{BAT}	PWR	---	---	---
D31	LED#	O	OD_{14}	V_{SB}	---
E1	FRAME#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
E2	PCICLK	I	IN_T	V_{IO}	---
E3	REQ1#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
E4	PCICLK1	O	O_{PCI}	V_{IO}	---
	LPC_ROM	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
E28	AV_{SSPLL3}	GND	---	---	---
E29	PWRBTN#	I (PU_{100})	IN_{BTN}	V_{SB}	---
E30[4, 5]	ONCTL#	O	OD_{14}	V_{SB}	---
E31	GPWIO0	I/O (PU_{100})	IN_{TS}, $TS_{2/14}$	V_{SB}	---

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
F1	IOR#	O	$O_{3/5}$	V_{IO}	PMR[21] = 0 and PMR[2] = 0
	DOCR#	O	$O_{3/5}$		PMR[21] = 0 and PMR[2] = 1
	GPIO14	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$		PMR[21] = 1 and PMR[2] = 1
F2	V_{SS}	GND	---	---	---
F3	RD#	O	$O_{3/5}$	V_{IO}	---
	CLKSEL0	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
F4	AD23	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A23	O	O_{PCI}		
F28	THRM#	I	IN_{TS}	V_{SB}	---
F29	V_{SB}	PWR	---	---	---
F30	V_{SS}	GND	---	---	---
F31[4, 5]	PWRCNT1	O	OD_{14}	V_{SB}	---
G1	WR#	O	$O_{3/5}$	V_{IO}	
G2	V_{IO}	PWR	---	---	---
G3	IOW#	O	$O_{3/5}$	V_{IO}	PMR[21] = 0 and PMR[2] = 0
	DOCW#	O	$O_{3/5}$		PMR[21] = 0 and PMR[2] = 1
	GPIO15	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$		PMR[21] = 1 and PMR[2] = 1
G4	ROMCS#	O	$O_{3/5}$	V_{IO}	---
	BOOT16	I (PD_{100})	IN_{STRP}	V_{IO}	Strap (See Table 2-6 on page 51.)
G28	GPWIO1	I/O (PU_{100})	IN_{Ts}, $TS_{2/14}$	V_{SB}	---
G29	GPWIO2	I/O (PU_{100})	IN_{TS}, $TS_{2/14}$	V_{SB}	---
G30	V_{IO}	PWR	---	---	---
G31[4, 5]	PWRCNT2	O	OD^{14}	V_{SB}	---
H1	TRDE#	O	$O_{3/5}$	V_{IO}	PMR[12] = 0
	GPIO0	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$	V_{IO}	PMR[12] = 1
H2	GPIO1	I/O ($PU_{22.5}$)	IN_{T}, $O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[13] = 0
	IOCS1#	O ($PU_{22.5}$)	$O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[13] = 1
	TFTD12	O ($PU_{22.5}$)	$O_{1/4}$	V_{IO}	PMR[23][3] = 1
H3	GPIO20	I/O ($PU_{22.5}$)	IN_{T}, $O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[7] = 0
	DOCCS#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[23][3] = 0 and PMR[7] = 1
	TFTD0	O ($PU_{22.5}$)	$O_{1/4}$		PMR[23][3] = 1
H4	GPIO19	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$	V_{IO}	PMR[9] = 0 and PMR[4] = 0
	INTC#	I ($PU_{22.5}$)	IN_{TS}		PMR[9] = 0 and PMR[4] = 1
	IOCHRDY	I ($PU_{22.5}$)	IN_{TS1}		PMR[9] = 1 and PMR[4] = 1
H28	V_{SBL}	PWR	---	---	---
H29	CLK32	O	$O_{2/5}$	V_{SB}	---
H30	GPIO11	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	RI2#	I ($PU_{22.5}$)	IN_{TS}		PMR[18] = 1 and PMR[8] = 0
	IRQ15	I ($PU_{22.5}$)	IN_{TS1}		PMR[18] = 0 and PMR[8] = 1
H31	SDATA_IN2	I	IN_{TS}	V_{SB}	F3BAR0+Memory Offset 08h[21] = 1
J1	HSYNC	O	$O_{1/4}$	V_{IO}	---
J2	VSYNC	O	$O_{1/4}$	V_{IO}	---
J3	IRTX	O	$O_{8/8}$	V_{IO}	PMR[6] = 0
	SOUT3	O	$O_{8/8}$		PMR[6] = 1
J4	GPIO17	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[5] = 0
	IOCS0#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[23][3] = 0 and PMR[5] = 1
	TFTDCK	O ($PU_{22.5}$)	$O_{1/4}$		PMR[23][3] = 1
J28	IRRX1	I	IN_{TS}	V_{SB}	PMR[6] = 0
	SIN3	I	IN_{TS}	V_{IO}	PMR[6] =1
J29	POR#	I	IN_{TS}	V_{IO}	---
J30[4]	MD0	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
J31[4]	MD1	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
K1	RED	O	WIRE	AV_{C-CCRT}	---
K2	V_{SSCRT}	GND	---	---	---
K3	V_{CCCRT}	PWR	---	---	---
K4	V_{SS}	GND	---	---	---
K28	V_{SS}	GND	---	---	---
K29[4]	MD2	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
K30[4]	MD3	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
K31[4]	MD4	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
L1	AV_{SSCRT}	GND	---	---	---
L2	V_{SS}	GND	---	---	---
L3	AV_{CCCRT}	PWR	---	---	---
L4	V_{CORE}	PWR	---	---	---
L28	V_{CORE}	PWR	---	---	---
L29[4]	MD5	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
L30	V_{SS}	GND	---	---	---
L31[4]	MD6	I/O	IN_{T}, $TS_{2/5}$	V_{IO}	---
M1	AV_{CCCRT}	PWR	---	---	---
M2	V_{IO}	PWR	---	---	---
M3	GREEN	O	WIRE	AV_{C-CCRT}	---

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
M4	V_{SS}	GND	---	---	---
M28	V_{SS}	GND	---	---	---
M29[4]	MD7	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
M30	V_{IO}	PWR	---	---	---
M31	DQM0	O	$O_{2/5}$	V_{IO}	---
N1	AV_{CCCRT}	PWR	---	---	---
N2	BLUE	O	WIRE	AV_{C-CCRT}	---
N3	AV_{SSCRT}	GND	---	---	---
N4	V_{CORE}	PWR	---	---	---
N28	V_{CORE}	PWR	---	---	---
N29	WEA#	O	$O_{2/5}$	V_{IO}	---
N30	CASA#	O	$O_{2/5}$	V_{IO}	---
N31	RASA#	O	$O_{2/5}$	V_{IO}	---
P1	VREF	I/O	WIRE	AV_{C-CCRT}	---
P2	SETRES	I	WIRE	AV_{C-CCRT}	---
P3	AV_{SSCRT}	GND	---	---	---
P4	V_{SS}	GND	---	---	---
P28	V_{SS}	GND	---	---	---
P29	CS0#	O	$O_{2/5}$	V_{IO}	---
P30	BA0	O	$O_{2/5}$	V_{IO}	---
P31	BA1	O	$O_{2/5}$	V_{IO}	---
R1	V_{PLL2}	PWR	---	---	---
R2	V_{SS}	GND	---	---	---
R3	AV_{SSPLL2}	GND	---	---	---
R4	V_{CORE}	PWR	---	---	---
R28	V_{CORE}	PWR	---	---	---
R29	MA10	O	$O_{2/5}$	V_{IO}	---
R30	V_{SS}	GND	---	---	---
R31	MA0	O	$O_{2/5}$	V_{IO}	---
T1[4, 5]	BUSY/WAIT#	I	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD3	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE1#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
T2	V_{IO}	PWR	---	---	---
T3[4, 5]	PE	I ($PU_{22.5}$ $PD_{22.5}$)	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0) (PU/PD under software control.)
	TFTD14	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE2#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
T4[4, 5]	SLCT	I	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD15	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE3#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
T28	DQM4	O	$O_{2/5}$	V_{IO}	---
T29	MA2	O	$O_{2/5}$	V_{IO}	---
T30	V_{CORE}	PWR	---	---	---
T31	MA1	O	$O_{2/5}$	V_{IO}	---
U1[4, 5]	PD7	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD13	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD7	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
U2	V_{SS}	GND	---	---	---
U3[4, 5]	ACK#	I	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTDE	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	FPCICLK	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
U4	V_{CORE}	PWR	---	---	---
U28	V_{CORE}	PWR	---	---	---
U29[4]	MD33	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
U30	V_{SS}	GND	---	---	---
U31[4]	MD32	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
V1[4, 5]	PD4	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD10	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD4	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
V2[4, 5]	PD5	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD11	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD5	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
V3[4, 5]	PD6	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD1	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0
	F_AD6	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
V4	V_{SS}	GND	---	---	---
V28	V_{SS}	GND	---	---	---
V29[4]	MD36	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
V30[4]	MD35	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
V31[4]	MD34	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
W1[4, 5]	SLIN#/ASTRB#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD16	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_IRDY#	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
W2[4, 5]	PD3	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD9	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD3	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
W3[4, 5]	PD2	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD8	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD2	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
W4	V_{CORE}	PWR	---	---	---
W28	V_{CORE}	PWR	---	---	---
W29[4]	MD39	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
W30[4]	MD38	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
W31[4]	MD37	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
Y1[4, 5]	PD1	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD7	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD1	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
Y2	V_{IO}	PWR	---	---	---
Y3[4, 5]	INIT#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD5	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	SMI_O	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
Y4	V_{SS}	GND	---	---	---
Y28	V_{SS}	GND	---	---	---
Y29[4]	MD46	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
Y30	V_{IO}	PWR		---	---
Y31[4]	MD47	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AA1[4, 5]	PD0	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD6	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD0	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
AA2	V_{SS}	GND	---	---	---
AA3[4, 5]	ERR#	I	IN_T, $O_{1/4}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD4	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE0#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
AA4	V_{CORE}	PWR	---	---	---
AA28	V_{CORE}	PWR	---	---	---
AA29[4]	MD44	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AA30	V_{SS}	GND	---	---	---
AA31[4]	MD45	I/O	IN_T, $TS_{2/5}$	V_{IO}	---

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AB1[4, 5]	STB#/WRITE#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD17	O	$O^{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_FRAME#	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
AB2[4, 5]	AFD#/DSTRB#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD2	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	INTR_O	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
AB3	NC	---	---	---	---
AB4	V_{SS}	GND	---	---	---
AB28	V_{SS}	GND	---	---	---
AB29[4]	MD41	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AB30[4]	MD42	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AB31[4]	MD43	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AC1	NC	---	---	---	---
AC2	NC	---	---	---	---
AC3	V_{IO}	PWR	---	---	---
AC4	V_{SS}	GND	---	---	---
AC28	CKEA	O	$O_{2/5}$	V_{IO}	---
AC29	SDCLK0	O	$O_{2/5}$	V_{IO}	---
AC30	DQM5	O	$O_{2/5}$	V_{IO}	---
AC31[4]	MD40	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AD1	NC	---	---	---	---
AD2	NC	---	---	---	---
AD3	NC	---	---	---	---
AD4	NC	---	---	---	---
AD28	MA6	O	$O_{2/5}$	V_{IO}	---
AD29	MA7	O	$O_{2/5}$	V_{IO}	---
AD30	MA8	O	$O_{2/5}$	V_{IO}	---
AD31	MA9	O	$O_{2/5}$	V_{IO}	---
AE1	NC	---	---	---	---
AE2	V_{IO}	PWR	---	---	---
AE3	INTA#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AE4[4]	DPOS_PORT3	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AE28	MA3	O	$O_{2/5}$	V_{IO}	---
AE29	MA4	O	$O_{2/5}$	V_{IO}	---
AE30	V_{IO}	PWR	---	---	---
AE31	MA5	O	$O_{2/5}$	V_{IO}	---
AF1	INTB#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AF2	V_{SS}	GND	---	---	---
AF3[4]	DNEG_PORT3	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AF4	AV_{CCUSB}	PWR	---	---	---
AF28[4]	MD14	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AF29[4]	MD15	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AF30	V_{SS}	GND	---	---	---
AF31	DQM1	O	$O_{2/5}$	V_{IO}	---
AG1	AV_{SSUSB}	GND	---	---	---
AG2[4]	DNEG_PORT2	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AG3[4]	DNEG_PORT1	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AG4	GPIO9	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	DCD2#	I ($PU_{22.5}$)	IN_{TS}		PMR[18] = 1 and PMR[8] = 0
	IDE_IOW1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 0 and PMR[8] = 1
	SDTEST2	O ($PU_{22.5}$)	$O_{2/5}$		PMR[18] = 1 and PMR[8] = 1
AG28	MA11	O	$O_{2/5}$	V_{IO}	---
AG29[4]	MD9	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AG30[4]	MD8	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AG31[4]	MD13	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH1[4]	DPOS_PORT2	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AH2[4]	DPOS_PORT1	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AH3	GPIO6	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	DTR2#/BOUT2	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 1 and PMR[8] = 0
	IDE_IOR1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 0 and PMR[8] = 1
	SDTEST5	O ($PU_{22.5}$)	$O_{2/5}$		PMR[18] = 1 and PMR[8] = 1

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AH4	GPIO7	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[17] = 0 and PMR[8] = 0
	RTS2#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[17] = 1 and PMR[8] = 0
	IDE_DACK1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[17] = 0 and PMR[8] = 1
	SDTEST0	O ($PU_{22.5}$)	$O_{2/5}$		PMR[17] = 1 and PMR[8] = 1
AH5	TDP	I/O	Diode	---	---
AH6	TDO	O	O_{PCI}	V_{IO}	---
AH7	VPCKIN	I	IN_T	V_{IO}	---
AH8	VPD4	I	IN_T	V_{IO}	---
AH9	VPD0	I	IN_T	V_{IO}	---
AH10	V_{SS}	GND	---	---	---
AH11	V_{CORE}	PWR	---	---	---
AH12	V_{SS}	GND	---	---	---
AH13	V_{CORE}	PWR	---	---	---
AH14	V_{SS}	GND	---	---	---
AH15	V_{CORE}	PWR	---	---	---
AH16	SDCLK1	O	$O_{2/5}$	V_{IO}	---
AH17	V_{CORE}	PWR	---	---	---
AH18	V_{SS}	GND	---	---	---
AH19	V_{CORE}	PWR	---	---	---
AH20	V_{SS}	GND	---	---	---
AH21	V_{CORE}	PWR	---	---	---
AH22	V_{SS}	GND	---	---	---
AH23[4]	MD28	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH24[4]	MD55	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH25[4]	MD51	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH26[4]	MD48	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH27[4]	MD23	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH28	SDCLK_OUT	O	$O_{2/5}$	V_{IO}	---
AH29	MA12	O	$O_{2/5}$	V_{IO}	---
AH30[4]	MD11	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH31[4]	MD10	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ1	GPIO10	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	DSR2#	I ($PU_{22.5}$)	IN_{TS}		PMR[18] = 1 and PMR[8] = 0
	IDE_IORDY1	I ($PU_{22.5}$)	IN_{TS1}		PMR[18] = 0 and PMR[8] = 1
	SDTEST1	O ($PU_{22.5}$)	$O_{2/5}$		PMR[18] = 1 and PMR[8] = 1
AJ2	GPIO8	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[17] = 0 and PMR[8] = 0
	CTS2#	I ($PU_{22.5}$)	IN_{TS}		PMR[17] = 1 and PMR[8] = 0
	IDE_DREQ1	I ($PU_{22.5}$)	IN_{TS1}		PMR[17] = 0 and PMR[8] = 1
	SDTEST4	O ($PU_{22.5}$)	$O_{2/5}$		PMR[17] = 1 and PMR[8] = 1
AJ3	V_{IO}	PWR	---	---	---
AJ4	SIN2	I	IN_{TS}	V_{IO}	PMR[28] = 0
	SDTEST3	O	$O_{2/5}$		PMR[28] = 1
AJ5	TMS	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AJ6	VPD7	I	IN_T	V_{IO}	---
AJ7	VPD6	I	IN_T	V_{IO}	---
AJ8	VPD2	I	IN_T	V_{IO}	---
AJ9	GPIO38/IRRX2	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0. The IRRX2 input is connected to the input path of GPIO38. There is no logic required to enable IRRX2, just a simple connection. Hence, when GPIO38 is the selected function, IRRX2 is also selected.
	LPCPD#	O	O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AJ10	GPIO35	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	LAD3	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AJ11	GPIO32	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	LAD0	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AJ12	GPIO12	I/O ($PU_{22.5}$)	IN_{AB}, $O_{8/8}$	V_{IO}	PMR[19] = 0
	AB2C	I/O ($PU_{22.5}$)	IN_{AB}, OD_8		PMR[19] = 1
AJ13	AB1C	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[23][3] = 0
	GPIO20	I/O ($PU_{22.5}$)	IN_T, $O_{3/5}$		PMR[23][3] = 1 and PMR[7] = 0
	DOCCS#	O	O3/5		PMR[23][3] = 1 and PMR[7] = 1
AJ14	AC97_CLK	O	$O_{2/5}$	V_{IO}	PMR[25] = 1
AJ15	AC97_RST#	O	$O_{2/5}$	V_{IO}	FPCI_MON = 0
	F_STOP#	O	$O_{2/5}$		FPCI_MON = 1
AJ16	SDCLK3	O	$O_{2/5}$	V_{IO}	---
AJ17[4]	MD56	I/O	IN_T, $TS_{2/5}$	V_{IO}	---

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AJ18[4]	MD58	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ19[4]	MD61	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ20	DQM7	O	$O_{2/5}$	V_{IO}	---
AJ21	DQM3	O	$O_{2/5}$	V_{IO}	---
AJ22[4]	MD25	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ23[4]	MD29	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ24[4]	MD54	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ25[4]	MD50	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ26	DQM6	O	$O_{2/5}$	V_{IO}	---
AJ27[4]	MD22	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ28[4]	MD19	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ29	V_{IO}	PWR	---	---	---
AJ30	SDCLK_IN	I	IN_T	V_{IO}	---
AJ31[4]	MD12	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK1	V_{IO}	PWR	---	---	---
AK2	V_{SS}	GND	---	---	---
AK3	SOUT2	O	$O_{8/8}$	V_{IO}	---
	CLKSEL2	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
AK4	TRST#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AK5	TDI	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AK6	V_{IO}	PWR	---	---	---
AK7	V_{SS}	GND	---	---	---
AK8	VPD1	I	IN_T	V_{IO}	---
AK9	GPIO37	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	LFRAME#	O	O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AK10	GPIO34	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	LAD2	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AK11	V_{IO}	PWR	---	---	---
AK12	V_{SS}	GND	---	---	---
AK13	SDATA_OUT	O	O_{AC97}	V_{IO}	---
	TFT_PRSNT	I (PD_{100})	IN_{STRP}	V_{IO}	Strap (See Table 2-6 on page 51.)
AK14	SDATA_IN	I	IN_T	V_{IO}	FPCI_MON = 0
	F_GNT0#	O	$O_{2/5}$		FPCI_MON = 1
AK15	V_{SS}	GND	---	---	---
AK16	V_{IO}	PWR	---	---	---

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AK17	V_{SS}	GND	---	---	---
AK18[4]	MD59	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK19[4]	MD62	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK20	V_{IO}	PWR	---	---	---
AK21	V_{SS}	GND	---	---	---
AK22[4]	MD26	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK23[4]	MD30	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK24[4]	MD53	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK25	V_{IO}	PWR	---	---	---
AK26	V_{SS}	GND	---	---	---
AK27[4]	MD21	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK28[4]	MD18	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK29	CS1#	O	$O_{2/5}$	V_{IO}	---
AK30	V_{SS}	GND	---	---	---
AK31	V_{IO}	PWR	---	---	---
AL1	V_{SS}	GND	---	---	---
AL2	V_{IO}	PWR	---	---	---
AL3	TDN	I/O	WIRE	V_{IO}	---
AL4	TCK	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AL5	GTEST	I ($PD_{22.5}$)	IN_T	V_{IO}	---
AL6	VPD5	I	IN_T	V_{IO}	---
AL7	VPD3	I	IN_T	V_{IO}	---
AL8	GPIO39	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	SERIRQ	I/O	IN_{PCI}, O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AL9	GPIO36	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	LDRQ#	I	IN_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AL10	GPIO33	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][6] = 0 and PMR[22][6] = 0
	LAD1	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][6] = 1 and PMR[22][6] = 1
AL11	GPIO13	I/O ($PU_{22.5}$)	IN_{AB}, $O_{8/8}$	V_{IO}	PMR[19] = 0
	AB2D	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[19] = 1
AL12	AB1D	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[23][3] = 0
	GPIO1	I/O ($PU_{22.5}$)	IN_T, $O_{3/5}$		PMR[23][3] = 1and PMR[13] = 0
	IOCS1#	O	$O_{3/5}$		PMR[23][3] = 1 and PMR[13] = 1

Signal Definitions (Continued)

Table 2-2. 432-EBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AL13	SYNC	O	O_{AC97}	V_{IO}	---
	CLKSEL3	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
AL14	BIT_CLK	I	IN_T	V_{IO}	FPCI_MON = 0
	F_TRDY#	O	$O_{1/4}$		FPCI_MON = 1
AL15	GPIO16	I/O ($PU_{22.5}$)	IN_T, $O_{2/5}$	V_{IO}	PMR[0] = 0 and FPCI_MON = 0
	PC_BEEP	O	$O_{2/5}$		PMR[0] = 1 = 0 and FPCI_MON = 0
	F_DEVSEL#	O	$O_{2/5}$		FPCI_MON = 1
AL16	GXCLK	O	$O_{2/5}$	V_{IO}	PMR[23][3] = 0 and PMR[29] = 0
	FP_VDD_ON	O	$O_{1/4}$		PMR[23][3] = 1
	TEST3	O	$O_{2/5}$		PMR[23][3] = 0 and PMR[29] = 1 and
AL17[4]	MD57	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL18[4]	MD60	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL19[4]	MD63	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL20	SDCLK2	O	$O_{2/5}$	V_{IO}	---
AL21[4]	MD24	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL22[4]	MD27	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL23[4]	MD31	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL24[4]	MD52	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL25[4]	MD49	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL26	DQM2	O	$O_{2/5}$	V_{IO}	---
AL27[4]	MD20	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL28[4]	MD17	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL29[4]	MD16	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL30	V_{IO}	PWR	---	---	---
AL31	V_{SS}	GND	---	---	---

1. For Buffer Type definitions, refer to Table 8-9 "Buffer Types" on page 365.
2. May need 5V tolerant protection at system level (DDC_SCL, DDC_SDA).
3. The TFT_PRSNT strap determines the power-on reset (POR) state of PMR[23].
4. Is back-drive protected (MD[63:0], DPOS_PORT1, DNEG_PORT1, DPOS_PORT2, DNEG_PORT2, DPOS_PORT3, DNEG_PORT3, ACK#, AFD#/DSTRB#, BUSY/WAIT#, ERR#, INIT#, PD[7:0], PE, SLCT, SLIN#/ASTRB#, STB#/WRITE#, ONCTL#, PWRCNT[2:1]).
5. Is 5V tolerant (ACK#, AFD#/DSTRB#, BUSY/WAIT#, ERR#, INIT#, PD[7:0], PE, SLCT, SLIN#/ASTRB#, STB#/WRITE#, ONCTL#, PWRCNT[2:1]).
6. The LPC_ROM strap determines the power-on reset (POR) state of PMR[14] and PMR[22].

Signal Definitions (Continued)

Table 2-3. 432-EBGA Ball Assignment - Sorted Alphabetically by Signal Name

Signal Name	Ball No.
A0	A17
A1	D16
A2	A18
A3	A15
A4	A16
A5	A14
A6	C15
A7	B14
A8	C14
A9	B13
A10	C13
A11	C12
A12	A12
A13	C11
A14	A11
A15	B10
A16	A7
A17	C7
A18	D7
A19	A6
A20	D6
A21	C6
A22	A5
A23	F4
AB1C	AJ13
AB1D	AL12
AB2C	AJ12
AB2D	AL11
AC97_CLK	AJ14
AC97_RST#	AJ15
ACK#	U3
AD0	A17
AD1	D16
AD2	A18
AD3	A15
AD4	A16
AD5	A14
AD6	C15
AD7	B14
AD8	C14
AD9	B13
AD10	C13
AD11	C12
AD12	A12
AD13	C11
AD14	A11
AD15	B10
AD16	A7
AD17	C7
AD18	D7
AD19	A6
AD20	D6
AD21	C6
AD22	A5
AD23	F4
AD24	C5
AD25	D5
AD26	A4
AD27	B4
AD28	C4
AD29	A3
AD30	C2
AD31	B3
AFD#/DSTRB#	AB2
AV_{CCCRT}	L3, M1, N1
AV_{CCUSB}	AF4
AV_{SSCRT}	L1, N3, P3
AV_{SSPLL2}	R3
AV_{SSPLL3}	E28
AV_{SSUSB}	AG1
BA0	P30
BA1	P31
BHE#	B5
BIT_CLK	AL14
BLUE	N2
BOOT16	G4
BUSY/WAIT#	T1
C/BE0#	A13
C/BE1#	A10
C/BE2#	D8
C/BE3#	A8
CASA#	N30
CKEA	AC28
CLK27M	A23
CLK32	H29
CLKSEL0	F3
CLKSEL1	B27
CLKSEL2	AK3
CLKSEL3	AL13
CS0#	P29
CS1#	AK29
CTS2#	AJ2
D0	C5
D1	D5
D2	A4
D3	B4
D4	C4
D5	A3
D6	C2
D7	B3
D8	A13
D9	A10
D10	D8
D11	A8
D12	C10
D13	B8
D14	C8
D15	D9
DCD2#	AG4
DDC_SCL	A20
DDC_SDA	C20
DEVSEL#	B5
DID0	D4
DID1	D2
DNEG_PORT1	AG3
DNEG_PORT2	AG2
DNEG_PORT3	AF3
DOCCS#	H3
DOCCS#	AJ13
DOCR#	F1
DOCW#	G3
DPOS_PORT1	AH2
DPOS_PORT2	AH1
DPOS_PORT3	AE4
DQM0	M31
DQM1	AF31
DQM2	AL26
DQM3	AJ21
DQM4	T28
DQM5	AC30
DQM6	AJ26
DQM7	AJ20
DSR2#	AJ1
DTR1#/BOUT1	A28
DTR2#/BOUT2	AH3
ERR#	AA3
F_AD0	AA1
F_AD1	Y1
F_AD2	W3
F_AD3	W2
F_AD4	V1
F_AD5	V2
F_AD6	V3
F_AD7	U1
F_C/BE0#	AA3
F_C/BE1#	T1

Signal Definitions (Continued)

Table 2-3. 432-EBGA Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
F_C/BE2#	T3
F_C/BE3#	T4
F_DEVSEL#	AL15
F_FRAME#	AB1
F_GNT0#	AK14
F_IRDY#	W1
F_STOP#	AJ15
F_TRDY#	AL14
FP_VDD_ON	B23, AL16
FPCI_MON	D3
FPCICLK	U3
FRAME#	E1
GNT0#	D4
GNT1#	D2
GPIO0	H1
GPIO1	H2, AL12
GPIO6	AH3
GPIO7	AH4
GPIO8	AJ2
GPIO9	AG4
GPIO10	AJ1
GPIO11	H30
GPIO12	AJ12
GPIO13	AL11
GPIO14	F1
GPIO15	G3
GPIO16	AL15
GPIO17	J4
GPIO18	A28
GPIO19	H4
GPIO20	H3, AJ13
GPIO32	AJ11
GPIO33	AL10
GPIO34	AK10
GPIO35	AJ10
GPIO36	AL9
GPIO37	AK9
GPIO38/IRRX2	AJ9
GPIO39	AL8
GPIO40	A21
GPIO41	C19
GPWIO0	E31
GPWIO1	G28
GPWIO2	G29
GREEN	M3
GTEST	AL5
GXCLK	AL16
HSYNC	J1
IDE_ADDR0	A26

Signal Name	Ball No.
IDE_ADDR1	C26
IDE_ADDR2	C17
IDE_CS0#	A27
IDE_CS1#	C16
IDE_DACK0#	C25
IDE_DACK1#	AH4
IDE_DATA0	B24
IDE_DATA1	A24
IDE_DATA2	D23
IDE_DATA3	C23
IDE_DATA4	B23
IDE_DATA5	A23
IDE_DATA6	C22
IDE_DATA7	B22
IDE_DATA8	A21
IDE_DATA9	C20
IDE_DATA10	A20
IDE_DATA11	C19
IDE_DATA12	B19
IDE_DATA13	A19
IDE_DATA14	C18
IDE_DATA15	B18
IDE_DREQ0	C24
IDE_DREQ1	AJ2
IDE_IOR0#	C21
IDE_IOR1#	AH3
IDE_IORDY0	A25
IDE_IORDY1	AJ1
IDE_IOW0#	D24
IDE_IOW1#	AG4
IDE_RST#	A22
INIT#	Y3
INTA#	AE3
INTB#	AF1
INTC#	H4
INTD#	B22
INTR_O	AB2
IOCHRDY	H4
IOCS0#	J4
IOCS1#	H2
IOCS1#	AL12
IOR#	F1
IOW#	G3
IRDY#	C8
IRQ9	C22
IRQ14	D25
IRQ15	H30
IRRX1	J28
IRTX	J3

Signal Name	Ball No.
LAD0	AJ11
LAD1	AL10
LAD2	AK10
LAD3	AJ10
LDRQ#	AL9
LED#	D31
LFRAME#	AK9
LOCK#	C9
LPC_ROM	E4
LPCPD#	AJ9
MA0	R31
MA1	T31
MA2	T29
MA3	AE28
MA4	AE29
MA5	AE31
MA6	AD28
MA7	AD29
MA8	AD30
MA9	AD31
MA10	R29
MA11	AG28
MA12	AH29
MD0	J30
MD1	J31
MD2	K29
MD3	K30
MD4	K31
MD5	L29
MD6	L31
MD7	M29
MD8	AG30
MD9	AG29
MD10	AH31
MD11	AH30
MD12	AJ31
MD13	AG31
MD14	AF28
MD15	AF29
MD16	AL29
MD17	AL28
MD18	AK28
MD19	AJ28
MD20	AL27
MD21	AK27
MD22	AJ27
MD23	AH27
MD24	AL21
MD25	AJ22

Signal Definitions (Continued)

Table 2-3. 432-EBGA Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
MD26	AK22
MD27	AL22
MD28	AH23
MD29	AJ23
MD30	AK23
MD31	AL23
MD32	U31
MD33	U29
MD34	V31
MD35	V30
MD36	V29
MD37	W31
MD38	W30
MD39	W29
MD40	AC31
MD41	AB29
MD42	AB30
MD43	AB31
MD44	AA29
MD45	AA31
MD46	Y29
MD47	Y31
MD48	AH26
MD49	AL25
MD50	AJ25
MD51	AH25
MD52	AL24
MD53	AK24
MD54	AJ24
MD55	AH24
MD56	AJ17
MD57	AL17
MD58	AJ18
MD59	AK18
MD60	AL18
MD61	AJ19
MD62	AK19
MD63	AL19
NC (Total of 8)	AB3, AC1, AC2, AD1, AD2, AD3, AD4, AE1
ONCTL#	E30
OVER_CUR#	C27
PAR	C10
PC_BEEP	AL15
PCICLK	E2
PCICLK0	D3
PCICLK1	E4
PCIRST#	D1

Signal Name	Ball No.
PD0	AA1
PD1	Y1
PD2	W3
PD3	W2
PD4	V1
PD5	V2
PD6	V3
PD7	U1
PE	T3
PERR#	B9
PLL2B	D28
PLL5B	B29
PLL6B	C28
POR#	J29
POWER_EN	B28
PWRBTN#	E29
PWRCNT1	F31
PWRCNT2	G31
RASA#	N31
RD#	F3
RED	K1
REQ0#	C1
REQ1#	E3
RI2#	H30
ROMCS#	G4
RTS2#	AH4
SDATA_IN	AK14
SDATA_IN2	H31
SDATA_OUT	AK13
SDCLK_IN	AJ30
SDCLK_OUT	AH28
SDCLK0	AC29
SDCLK1	AH16
SDCLK2	AL20
SDCLK3	AJ16
SDTEST0	AH4
SDTEST1	AJ1
SDTEST2	AG4
SDTEST3	AJ4
SDTEST4	AJ2
SDTEST5	AH3
SERIRQ	AL8
SERR#	A9
SETRES	P2
SIN1	D26
SIN2	AJ4
SIN3	J28
SLCT	T4
SLIN#/ASTRB#	W1

Signal Name	Ball No.
SMI_O	Y3
SOUT1	B27
SOUT2	AK3
SOUT3	J3
STB#/WRITE#	AB1
STOP#	D9
SYNC	AL13
TCK	AL4
TDI	AK5
TDN	AL3
TDO	AH6
TDP	AH5
TEST0	D28
TEST1	C28
TEST2	B29
TEST3	AL16
TFT_PRSNT	AK13
TFTD0	C25, H3
TFTD1	D25, V3
TFTD10	C21, V1
TFTD11	A25, V2
TFTD12	C23, H2
TFTD13	B19, U1
TFTD14	D23, T3
TFTD15	A19, T4
TFTD16	A24, W1
TFTD17	C18, AB1
TFTD2	C26, AB2
TFTD3	A26, T1
TFTD4	C17, AA3
TFTD5	A27, Y3
TFTD6	B24, AA1
TFTD7	B18, Y1
TFTD8	C24, W3
TFTD9	D24, W2
TFTDCK	A22, J4
TFTDE	C16, U3
THRM#	F28
TMS	AJ5
TRDE#	H1
TRDY#	B8
TRST#	AK4
V_{BAT}	D30
V_{CCCRT}	K3

Signal Definitions (Continued)

Table 2-3. 432-EBGA Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
V_{CORE} (Total of 25)	D11, D13, D15, D17, D19, D21, L4, L28, N4, N28, R4, R28, T30, U4, U28, W4, W28, AA4, AA28, AH11, AH13, AH15, AH17, AH19, AH21
V_{IO} (Total of 32)	A2, A30, B1, B6, B11, B16, B20, B25, B31, C3, C29, G2, G30, M2, M30, T2, Y2, Y30, AC3, AE2, AE30, AJ3, AJ29, AK1, AK6, AK11, AK16, AK20, AK25, AK31, AL2, AL30
VPCKIN	AH7
VPD0	AH9
VPD1	AK8
VPD2	AJ8
VPD3	AL7
VPD4	AH8
VPD5	AL6
VPD6	AJ7
VPD7	AJ6
V_{PLL2}	R1
V_{PLL3}	C31

Signal Name	Ball No.
VREF	P1
V_{SB}	F29
V_{SBL}	H28
V_{SS} (Total of 57)	A1, A31, B2, B7, B12, B15, B17, B21, B26, B30, D10, D12, D14, D18, D20, D22, F2, F30, K4, K28, L2, L30, M4, M28, P4, P28, R2, R30, U2, U30, V4, V28, Y4, Y28, AA2, AA30, AB4, AB28, AC4, AF2, AF30, AH10, AH12, AH14, AH18, AH20, AH22, AK2, AK7, AK12, AK15, AK17, AK21, AK26, AK30, AL1, AL31,
V_{SSCRT}	K2
VSYNC	J2
WEA#	N29
WR#	G1
X27I	A29
X27O	D27
X32I	C30
X32O	D29

Signal Definitions (Continued)



Note: Signal names have been abbreviated in this figure due to space constraints.

- ● = GND Ball
- ○ = PWR Ball
- Ⓢ = Strap Option Ball
- ⊕ = Multiplexed Ball

Figure 2-3. 481-TEPBGA Ball Assignment Diagram

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
A1	V_{SS}	GND	---	---	---
A2	V_{IO}	PWR	---	---	---
A3	AD30	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D6	I/O	IN_{PCI}, O_{PCI}		
A4	PCICLK0	O	O_{PCI}	V_{IO}	---
	FPCI_MON	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
A5	REQ1#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
A6	PCIRST#	O	O_{PCI}	V_{IO}	---
A7	PCICLK	I	IN_T	V_{IO}	---
A8	IOW#	O	$O_{3/5}$	V_{IO}	PMR[21] = 0 and PMR[2] = 0
	DOCW#	O	$O_{3/5}$		PMR[21] = 0 and PMR[2] = 1
	GPIO15	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$		PMR[21] = 1 and PMR[2] = 1
A9	GPIO20	I/O ($PU_{22.5}$)	IN_T, $O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[7] = 0
	DOCCS#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[23][3] = 0 and PMR[7] = 1
	TFTD0	O ($PU_{22.5}$)	$O_{1/4}$		PMR[23][3] = 1
A10	GPIO17	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[5] = 0
	IOCS0#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[23][3] = 0 and PMR[5] = 1
	TFTDCK	O ($PU_{22.5}$)	$O_{1/4}$		PMR[23][3] = 1
A11	HSYNC	O	$O_{1/4}$	V_{IO}	---
A12	AV_{CCCRT}	PWR	---	---	---
A13	V_{SS}	GND	---	---	---
A14	GREEN	O	WIRE	AV_{C-CCRT}	---
A15	BLUE	O	WIRE	AV_{C-CCRT}	---
A16	V_{SS}	GND	---	---	---
A17	V_{PLL2}	PWR	---	---	---
A18[6, 2]	PD7	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD13	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD7	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
A19	V_{SS}	GND	---	---	---
A20[6, 2]	PD6	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD1	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0
	F_AD6	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
A21[6, 2]	PD1	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD7	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD1	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
A22[6, 2]	STB#/WRITE#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD17	O	$O^{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_FRAME#	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
A23	NC	---	---	---	---
A24	NC	---	---	---	---
A25	NC	---	---	---	---
A26[6]	DPOS_PORT3	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
A27[6]	DNEG_PORT3	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
A28[6]	DPOS_PORT1	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
A29[6]	DNEG_PORT1	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
A30	V_{IO}	PWR	---	---	---
A31	V_{SS}	GND	---	---	---
B1	V_{SS}	GND	---	---	---
B2	V_{IO}	PWR	---	---	---
B3	AD29	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D5	I/O	IN_{PCI}, O_{PCI}		
B4	AD28	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D4	I/O	IN_{PCI}, O_{PCI}		
B5	REQ0#	I ($PU_{22.5}$)	INPCI	V_{IO}	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
B6	AD23	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A23	O	O_{PCI}		
B7	V_{SS}	GND	---	---	---
B8	RD#	O	$O_{3/5}$	V_{IO}	---
	CLKSEL0	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
B9	WR#	O	$O_{3/5}$	V_{IO}	
B10	V_{SS}	GND	---	---	---
B11	VSYNC	O	$O_{1/4}$	V_{IO}	---
B12	RED	O	WIRE	AV_{C-CCRT}	---
B13	V_{IO}	PWR	---	---	---
B14	AV_{SSCRT}	GND	---	---	---
B15	SETRES	I	WIRE	AV_{C-CCRT}	---
B16	V_{IO}	PWR	---	---	---
B17[6, 2]	BUSY/WAIT#	I	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD3	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE1#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
B18[6, 2]	ACK#	I	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTDE	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	FPCICLK	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
B19	V_{IO}	PWR	---	---	---
B20[6,2]	SLIN#/ASTRB#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD16	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_IRDY#	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
B21[6,2]	INIT#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD5	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	SMI_O	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
B22	V_{SS}	GND	---	---	---
B23	NC	---	---	---	---
B24	V_{SS}	GND	---	---	---
B25	V_{SS}	GND	---	---	---
B26	NC	---	---	---	---
B27[6]	DPOS_PORT2	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
B28[6]	DNEG_PORT2	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
B29	GPIO10	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	DSR2#	I ($PU_{22.5}$)	IN_{TS}		PMR[18] = 1 and PMR[8] = 0
	IDE_IORDY1	I ($PU_{22.5}$)	IN_{TS1}		PMR[18] = 0 and PMR[8] = 1
	SDTEST1	O ($PU_{22.5}$)	$O_{2/5}$		PMR[18] = 1 and PMR[8] = 1
B30	V_{SS}	GND	---	---	---
B31	V_{IO}	PWR	---	---	---
C1	AD26	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D2	I/O	IN_{PCI}, O_{PCI}		
C2	AD24	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D0	I/O	IN_{PCI}, O_{PCI}		
C3	V_{IO}	PWR	---	---	---
C4	AD25	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D1	I/O	IN_{PCI}, O_{PCI}		
C5	GNT0#	O	O_{PCI}	V_{IO}	---
	DID0	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
C6	GNT1#	O	O_{PCI}	V_{IO}	---
	DID1	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
C7	V_{IO}	PWR	---	---	---
C8	ROMCS#	O	$O_{3/5}$	V_{IO}	---
	BOOT16	I (PD_{100})	IN_{STRP}	V_{IO}	Strap (See Table 2-6 on page 51.)
C9	GPIO19	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$	V_{IO}	PMR[9] = 0 and PMR[4] = 0
	INTC#	I ($PU_{22.5}$)	IN_{TS}		PMR[9] = 0 and PMR[4] = 1
	IOCHRDY	I ($PU_{22.5}$)	IN_{TS1}		PMR[9] = 1 and PMR[4] = 1
C10	V_{IO}	PWR	---	---	---
C11	IRTX	O	$O_{8/8}$	V_{IO}	PMR[6] = 0
	SOUT3	O	$O_{8/8}$		PMR[6] = 1
C12	V_{SSCRT}	GND	---	---	---
C13	AV_{CCCRT}	PWR	---	---	---
C14	AV_{SSCRT}	GND	---	---	---
C15	AV_{SSCRT}	GND	---	---	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
C16	AV_{SSPLL2}	GND	---	---	---
C17[6,2]	SLCT	I	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD15	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE3#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
C18	PD4	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD10	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD4	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
C19[6,2]	PD5	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD11	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD5	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
C20[6,2]	PD3	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD9	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD3	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
C21[6,2]	PD0	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD6	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD0	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
C22	V_{IO}	PWR	---	---	---
C23	NC	---	---	---	---
C24	NC	---	---	---	---
C25	V_{IO}	PWR	---	---	---
C26	INTB#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
C27	AV_{SSUSB}	GND	---	---	---
C28	GPIO9	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	DCD2#	I ($PU_{22.5}$)	IN_{TS}		PMR[18] = 1 and PMR[8] = 0
	IDE_IOW1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 0 and PMR[8] = 1
	SDTEST2	O ($PU_{22.5}$)	$O_{2/5}$		PMR[18] = 1 and PMR[8] = 1
C29	V_{IO}	PWR	---	---	---
C30	GPIO7	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[17] = 0 and PMR[8] = 0
	RTS2#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[17] = 1 and PMR[8] = 0
	IDE_DACK1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[17] = 0 and PMR[8] = 1
	SDTEST0	O ($PU_{22.5}$)	$O_{2/5}$		PMR[17] = 1 and PMR[8] = 1
C31	GPIO8	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[17] = 0 and PMR[8] = 0
	CTS2#	I ($PU_{22.5}$)	IN_{TS}		PMR[17] = 1 and PMR[8] = 0
	IDE_DREQ1	I ($PU_{22.5}$)	IN_{TS1}		PMR[17] = 0 and PMR[8] = 1
	SDTEST4	O ($PU_{22.5}$)	$O_{2/5}$		PMR[17] = 1 and PMR[8] = 1
D1	AD21	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A21	O	O_{PCI}		
D2	AD22	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A22	O	O_{PCI}		
D3	AD20	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A20	O	O_{PCI}		
D4	AD27	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D3	I/O	IN_{PCI}, O_{PCI}		
D5	AD31	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D7	I/O	IN_{PCI}, O_{PCI}		
D6	PCICLK1	O	O_{PCI}	V_{IO}	---
	LPC_ROM	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
D7	V_{SS}	GND	---	---	---
D8	FRAME#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
D9	IOR#	O	$O_{3/5}$	V_{IO}	PMR[21] = 0 and PMR[2] = 0
	DOCR#	O	$O_{3/5}$		PMR[21] = 0 and PMR[2] = 1
	GPIO14	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$		PMR[21] = 1 and PMR[2] = 1
D10	GPIO1	I/O ($PU_{22.5}$)	IN_T, $O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[13] = 0
	IOCS1#	O ($PU_{22.5}$)	$O_{3/5}$	V_{IO}	PMR[23][3] = 0 and PMR[13] = 1
	TFTD12	O ($PU_{22.5}$)	$O_{1/4}$	V_{IO}	PMR[23][3] = 1
D11	TRDE#	O	$O_{3/5}$	V_{IO}	PMR[12] = 0
	GPIO0	I/O ($PU_{22.5}$)	IN_{TS}, $O_{3/5}$	V_{IO}	PMR[12] = 1
D12	V_{CCCRT}	PWR	---	---	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
D13	V_{SS}	GND	---	---	---
D14	V_{IO}	PWR	---	---	---
D15	AV_{CCCRT}	PWR	---	---	---
D16	VREF	I/O	WIRE	AV_{C-CCRT}	---
D17[6, 2]	PE	I ($PU_{22.5}$ $PD_{22.5}$)	IN_T	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0) (PU/PD under software control.)
	TFTD14	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE2#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
D18	V_{IO}	PWR	---	---	---
D19	V_{SS}	GND	---	---	---
D20[6, 2]	PD2	I/O	IN_T, $O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD8	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_AD2	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
D21[6, 2]	ERR#	I	IN_T, $O_{1/4}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD4	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	F_C/BE0#	O	$O_{1/4}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
D22[6, 2]	AFD#/DSTRB#	O	$O_{14/14}$	V_{IO}	PMR[23][3] = 0 and (PMR[27] = 0 and FPCI_MON = 0)
	TFTD2	O	$O_{1/4}$		PMR[23][3] = 1 and (PMR[27] = 0 and FPCI_MON = 0)
	INTR_O	O	$O_{14/14}$		PMR[23][3] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
D23	V_{IO}	PWR	---	---	---
D24	NC	---	---	---	---
D25	V_{SS}	GND	---	---	---
D26	INTA#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
D27	AV_{CCUSB}	PWR	---	---	---
D28	GPIO6	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	DTR2#/BOUT2	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 1 and PMR[8] = 0
	IDE_IOR1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 0 and PMR[8] = 1
	SDTEST5	O ($PU_{22.5}$)	$O_{2/5}$		PMR[18] = 1 and PMR[8] = 1
D29	SOUT2	O	$O_{8/8}$	V_{IO}	---
	CLKSEL2	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
D30	TDP	I/O	Diode	---	---
D31	TDN	I/O	WIRE	V_{IO}	---
E1	AD16	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A16	O	O_{PCI}		
E2	AD19	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A19	O	O_{PCI}		
E3	AD18	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A18	O	O_{PCI}		
E4	DEVSEL#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	BHE#	O	O_{PCI}		
E28	SIN2	I	IN_{TS}	V_{IO}	PMR[28] = 0
	SDTEST3	O	$O_{2/5}$		PMR[28] = 1
E29	TRST#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
E30	TDO	O	O_{PCI}	V_{IO}	---
E31	TCK	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
F1	TRDY#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D13	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
F2	IRDY#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D14	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
F3	C/BE2#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D10	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
F4	AD17	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A17	O	O_{PCI}		
F28	TMS	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
F29	TDI	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
F30	GTEST	I ($PD_{22.5}$)	IN_T	V_{IO}	---
F31	VPCKIN	I	IN_T	V_{IO}	---
G1	STOP#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D15	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
G2	V_{SS}	GND	---	---	---
G3	V_{IO}	PWR	---	---	---
G4	V_{SS}	GND	---	---	---
G28	V_{SS}	GND	---	---	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
G29	V_{IO}	PWR	---	---	---
G30	V_{SS}	GND	---	---	---
G31	VPD7	I	IN_T	V_{IO}	---
H1	SERR#	I/O ($PU_{22.5}$)	IN_{PCI}, OD_{PCI}	V_{IO}	---
H2	PERR#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
H3	LOCK#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
H4	C/BE3#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D11	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
H28	VPD6	I	IN_T	V_{IO}	---
H29	VPD5	I	IN_T	V_{IO}	---
H30	VPD4	I	IN_T	V_{IO}	---
H31	VPD3	I	IN_T	V_{IO}	---
J1	AD13	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A13	O	O_{PCI}		
J2	C/BE1#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D9	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
J3	AD15	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A15	O	O_{PCI}		
J4	PAR	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D12	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
J28	VPD2	I	IN_T	V_{IO}	---
J29	VPD1	I	IN_T	V_{IO}	---
J30	VPD0	I	IN_T	V_{IO}	---
J31	GPIO39	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	SERIRQ	I/O	IN_{PCI}, O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
K1	AD11	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A11	O	O_{PCI}		
K2	V_{IO}	PWR	---	---	---
K3	V_{SS}	GND	---	---	---
K4	AD14	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A14	O	O_{PCI}		
K28	GPIO38/IRRX2	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0. The IRRX2 input is connected to the input path of GPIO38. There is no logic required to enable IRRX2, just a simple connection. Hence, when GPIO38 is the selected function, IRRX2 is also selected.
	LPCPD#	O	O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
K29	V_{IO}	PWR	---	---	---
K30	V_{SS}	GND	---	---	---
K31	GPIO37	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	LFRAME#	O	O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
L1	C/BE0#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D8	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
L2	AD9	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A9	O	O_{PCI}		
L3	AD10	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A10	O	O_{PCI}		
L4	AD12	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A12	O	O_{PCI}		
L28	GPIO36	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	LDRQ#	I	IN_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
L29	GPIO35	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	LAD3	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
L30	GPIO34	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	LAD2	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
L31	GPIO33	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	LAD1	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
M1	V_{SS}	GND	---	---	---
M2	AD7	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A7	O	O_{PCI}		
M3	V_{IO}	PWR	---	---	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
M4	AD8	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A8	O	O_{PCI}		
M28	GPIO32	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14][4] = 0 and PMR[22][4] = 0
	LAD0	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14][4] = 1 and PMR[22][4] = 1
M29	GPIO13	I/O ($PU_{22.5}$)	IN_{AB}, $O_{8/8}$	V_{IO}	PMR[19] = 0
	AB2D	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[19] = 1
M30	V_{IO}	PWR	---	---	---
M31	V_{SS}	GND	---	---	---
N1	AD3	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A3	O	O_{PCI}		
N2	AD6	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A6	O	O^{PCI}		
N3	AD5	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A5	O	O_{PCI}		
N4	V_{SS}	GND	---	---	---
N13	V_{CORE}	PWR	---	---	---
N14	V_{CORE}	PWR	---	---	---
N15	V_{SS}	GND	---	---	---
N16	V_{SS}	GND	---	---	---
N17	V_{SS}	GND	---	---	---
N18	V_{CORE}	PWR	---	---	---
N19	V_{CORE}	PWR	---	---	---
N28	V_{SS}	GND	---	---	---
N29	GPIO12	I/O ($PU_{22.5}$)	IN_{AB}, $O_{8/8}$	V_{IO}	PMR[19] = 0
	AB2C	I/O ($PU_{22.5}$)	IN_{AB}, OD_8		PMR[19] = 1
N30	AB1D	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[23][3] = 0
	GPIO1	I/O ($PU_{22.5}$)	IN_T, $O_{3/5}$		PMR[23][3] = 1 and PMR[13] = 0
	IOCS1#	O	$O_{3/5}$		PMR[23][3] = 1 and PMR[13] = 1
N31	AB1C	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[23][3] = 0
	GPIO20	I/O ($PU_{22.5}$)	IN_T, $O_{3/5}$		PMR[23][3] = 1 and PMR[7] = 0
	DOCCS#	O	O3/5		PMR[23][3] = 1 and PMR[7] = 1
P1	AD4	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A4	O	O_{PCI}		
P2	IDE_CS1#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTDE	O	$O_{1/4}$		PMR[24] = 1
P3	AD1	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A1	O	O_{PCI}		
P4	V_{CORE}	PWR	---	---	---
P13	V_{CORE}	PWR	---	---	---
P14	V_{CORE}	PWR	---	---	---
P15	V_{SS}	GND	---	---	---
P16	V_{SS}	GND	---	---	---
P17	V_{SS}	GND	---	---	---
P18	V_{CORE}	PWR	---	---	---
P19	V_{CORE}	PWR	---	---	---
P28	V_{CORE}	PWR	---	---	---
P29	SDATA_OUT	O	O_{AC97}	V_{IO}	---
	TFT_PRSNT	I (PD_{100})	IN_{STRP}	V_{IO}	Strap (See Table 2-6 on page 51.)
P30	SYNC	O	O_{AC97}	V_{IO}	---
	CLKSEL3	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
P31	AC97_CLK	O	$O_{2/5}$	V_{IO}	PMR[25] = 1
R1	V_{SS}	GND	---	---	---
R2	V_{SS}	GND	---	---	---
R3	V_{SS}	GND	---	---	---
R4	V_{SS}	GND	---	---	---
R13	V_{SS}	GND	---	---	---
R14	V_{SS}	GND	---	---	---
R15	V_{SS}	GND	---	---	---
R16	V_{SS}	GND	---	---	---
R17	V_{SS}	GND	---	---	---
R18	V_{SS}	GND	---	---	---
R19	V_{SS}	GND	---	---	---
R28	V_{SS}	GND	---	---	---
R29	V_{SS}	GND	---	---	---
R30	V_{SS}	GND	---	---	---
R31	V_{SS}	GND	---	---	---
T1	V_{CORE}	PWR	---	---	---
T2	V_{CORE}	PWR	---	---	---
T3	V_{CORE}	PWR	---	---	---
T4	V_{CORE}	PWR	---	---	---
T13	V_{SS}	GND	---	---	---
T14	V_{SS}	GND	---	---	---
T15	V_{SS}	GND	---	---	---
T16	V_{SS}	GND	---	---	---
T17	V_{SS}	GND	---	---	---
T18	V_{SS}	GND	---	---	---
T19	V_{SS}	GND	---	---	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
T28	V_{CORE}	PWR	---	---	---
T29	V_{CORE}	PWR	---	---	---
T30	V_{CORE}	PWR	---	---	---
T31	V_{CORE}	PWR	---	---	---
U1	AD0	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A0	O	O_{PCI}		
U2	IDE_ADDR2	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD4	O	$O_{1/4}$		PMR[24] = 1
U3	AD2	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A2	O	O_{PCI}		
U4	V_{CORE}	PWR	---	---	---
U13	V_{SS}	GND	---	---	---
U14	V_{SS}	GND	---	---	---
U15	V_{SS}	GND	---	---	---
U16	V_{SS}	GND	---	---	---
U17	V_{SS}	GND	---	---	---
U18	V_{SS}	GND	---	---	---
U19	V_{SS}	GND	---	---	---
U28	V_{CORE}	PWR	---	---	---
U29	AC97_RST#	O	$O_{2/5}$	V_{IO}	FPCI_MON = 0
	F_STOP#	O	$O_{2/5}$		FPCI_MON = 1
U30	BIT_CLK	I	IN_T	V_{IO}	FPCI_MON = 0
	F_TRDY#	O	$O_{1/4}$		FPCI_MON = 1
U31	SDATA_IN	I	IN_T	V_{IO}	FPCI_MON = 0
	F_GNT0#	O	$O_{2/5}$		FPCI_MON = 1
V1	IDE_DATA15	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD7	O	$O_{1/4}$		PMR[24] = 1
V2	IDE_DATA14	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD17	O	$O_{1/4}$		PMR[24] = 1
V3	IDE_DATA13	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD15	O	$O_{1/4}$		PMR[24] = 1
V4	V_{SS}	GND	---	---	---
V13	V_{CORE}	PWR	---	---	---
V14	V_{CORE}	PWR	---	---	---
V15	V_{SS}	GND	---	---	---
V16	V_{SS}	GND	---	---	---
V17	V_{SS}	GND	---	---	---
V18	V_{CORE}	PWR	---	---	---
V19	V_{CORE}	PWR	---	---	---
V28	V_{SS}	GND	---	---	---
V29	SDCLK3	O	$O_{2/5}$	V_{IO}	---
V30	GXCLK	O	$O_{2/5}$	V_{IO}	PMR[23][3] = 0 and PMR[29] = 0
	FP_VDD_ON	O	$O_{1/4}$		PMR[23][3] = 1
	TEST3	O	$O_{2/5}$		PMR[23][3] = 0 and PMR[29] = 1
V31	GPIO16	I/O ($PU_{22.5}$)	IN_T, $O_{2/5}$	V_{IO}	PMR[0] = 0 and FPCI_MON = 0
	PC_BEEP	O	$O_{2/5}$		PMR[0] = 1 = 0 and FPCI_MON = 0
	F_DEVSEL#	O	$O_{2/5}$		FPCI_MON = 1
W1	V_{IO}	PWR	---	---	---
W2	V_{SS}	GND	---	---	---
W3	IDE_DATA12	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD13	O	$O_{1/4}$		PMR[24] = 1
W4	IDE_DATA11	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	GPIO41	I/O	IN_{TS1}, $O_{1/4}$		PMR[24] = 1
W13	V_{CORE}	PWR	---	---	---
W14	V_{CORE}	PWR	---	---	---
W15	V_{SS}	GND	---	---	---
W16	V_{SS}	GND	---	---	---
W17	V_{SS}	GND	---	---	---
W18	V_{CORE}	PWR	---	---	---
W19	V_{CORE}	PWR	---	---	---
W28[6]	MD57	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
W29	SDCLK1	O	$O_{2/5}$	V_{IO}	---
W30	V_{SS}	GND	---	---	---
W31	V_{IO}	PWR	---	---	---
Y1[5]	IDE_DATA10	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	DDC_SCL	O	OD_4		PMR[24] = 1
Y2[5]	IDE_DATA9	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	DDC_SDA	I/O	IN_T, OD_4		PMR[24] = 1
Y3	IDE_DATA8	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	GPIO40	I/O	IN_{TS1}, $O_{1/4}$		PMR[24] = 1
Y4	IDE_IOR0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD10	O	$O_{1/4}$		PMR[24] = 1
Y28[6]	MD58	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
Y29[6]	MD59	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
Y30[6]	MD60	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
Y31[6]	MD56	I/O	IN_T, $TS_{2/5}$	V_{IO}	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AA1	IDE_RST#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTDCK	O	$O_{1/4}$		PMR[24] = 1
AA2	IDE_DATA7	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	INTD#	I	IN_{TS}		PMR[24] = 1
AA3	IDE_DATA6	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	IRQ9	I	IN_{TS1}		PMR[24] = 1
AA4	IDE_DATA5	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	CLK27M	O	$O_{1/4}$		PMR[24] = 1
AA28	SDCLK2	O	$O_{2/5}$	V_{IO}	---
AA29[6]	MD61	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AA30[6]	MD62	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AA31[6]	MD63	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AB1	IDE_DATA4	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	FP_VDD_ON	O	$O_{1/4}$		PMR[24] = 1
AB2	V_{SS}	GND	---	---	---
AB3	V_{IO}	PWR	---	---	---
AB4	IDE_DATA3	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD12	O	$O_{1/4}$		PMR[24] = 1
AB28[6]	MD24	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AB29	V_{IO}	PWR	---	---	---
AB30	V_{SS}	GND	---	---	---
AB31	DQM7	O	$O_{2/5}$	V_{IO}	---
AC1	IDE_DATA1	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD16	O	$O_{1/4}$		PMR[24] = 1
AC2	IDE_DATA2	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD14	O	$O_{1/4}$		PMR[24] = 1
AC3	IDE_DATA0	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD6	O	$O_{1/4}$		PMR[24] = 1
AC4	IDE_DREQ0	I	IN_{TS1}	V_{IO}	PMR[24] = 0
	TFTD8	O	$O_{1/4}$		PMR[24] = 1
AC28[6]	MD25	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AC29[6]	MD26	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AC30[6]	MD27	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AC31	DQM3	O	$O_{2/5}$	V_{IO}	---
AD1	IDE_IORDY0	I	IN_{TS1}	V_{IO}	PMR[24] = 0
	TFTD11	O	$O_{1/4}$		PMR[24] = 1
AD2	IDE_IOW0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD9	O	$O_{1/4}$		PMR[24] = 1
AD3	IDE_ADDR0	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD3	O	$O_{1/4}$		PMR[24] = 1
AD4	IDE_DACK0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD0	O	$O_{1/4}$		PMR[24] = 1
AD28[6]	MD52	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AD29[6]	MD29	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AD30[6]	MD30	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AD31[6]	MD31	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AE1	IDE_ADDR1	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD2	O	$O_{1/4}$		PMR[24] = 1
AE2	V_{SS}	GND	---	---	---
AE3	V_{IO}	PWR	---	---	---
AE4	V_{SS}	GND	---	---	---
AE28	V_{SS}	GND	---	---	---
AE29	V_{IO}	PWR	---	---	---
AE30	V_{SS}	GND	---	---	---
AE31[6]	MD28	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AF1	IRQ14	I	IN_{TS1}	V_{IO}	PMR[24] = 0
	TFTD1	O	$O_{1/4}$		PMR[24] = 1
AF2	IDE_CS0#	O	$O_{1/4}$	V_{IO}	PMR[24] = 0
	TFTD5	O	$O_{1/4}$		PMR[24] = 1
AF3	SOUT1	O	$O_{8/8}$	V_{IO}	---
	CLKSEL1	I (PD_{100})	IN_{STRP}		Strap (See Table 2-6 on page 51.)
AF4	OVER_CUR#	I	IN_{TS}	V_{IO}	---
AF28[6]	MD50	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AF29[6]	MD49	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AF30[6]	MD54	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AF31[6]	MD53	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AG1	GPIO18	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[16] = 0
	DTR1#/BOUT1	O ($PU_{22.5}$)	$O_{8/8}$		PMR[16] =1
AG2	SIN1	I	IN_{TS}	V_{IO}	---
AG3	X27I	I	WIRE	V_{IO}	---
AG4	PLL6B	I/O	IN_{TS}, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST1	O	$O_{2/5}$		PMR[29] = 1
AG28[6]	MD21	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AG29	DQM6	O	$O_{2/5}$	V_{IO}	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AG30	DQM2	O	$O_{2/5}$	V_{IO}	---
AG31[6]	MD55	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH1	POWER_EN	O	$O_{1/4}$	V_{IO}	---
AH2	X27O	O	WIRE	V_{IO}	---
AH3	PLL2B	I/O	IN_T, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST0	O	$O_{2/5}$		PMR[29] = 1
AH4	V_{IO}	PWR	---	---	---
AH5	PWRBTN#	I (PU_{100})	IN_{BTN}	V_{SB}	---
AH6	GPWIO0	I/O (PU_{100})	IN_{TS}, $TS_{2/14}$	V_{SB}	---
AH7	V_{SS}	GND	---	---	---
AH8	CLK32	O	$O_{2/5}$	V_{SB}	---
AH9	POR#	I	IN_{TS}	V_{IO}	---
AH10[6]	MD3	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH11[6]	MD5	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH12	WEA#	O	$O_{2/5}$	V_{IO}	---
AH13	V_{SS}	GND	---	---	---
AH14	V_{IO}	PWR	---	---	---
AH15	MA1	O	$O_{2/5}$	V_{IO}	---
AH16[6]	MD34	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH17[6]	MD37	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH18	V_{IO}	PWR	---	---	---
AH19	V_{SS}	GND	---	---	---
AH20[6]	MD41	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH21	MA9	O	$O_{2/5}$	V_{IO}	---
AH22	MA8	O	$O_{2/5}$	V_{IO}	---
AH23	DQM1	O	$O_{2/5}$	V_{IO}	---
AH24[6]	MD13	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH25	V_{SS}	GND	---	---	---
AH26	MA11	O	$O_{2/5}$	V_{IO}	---
AH27	CS1#	O	$O_{2/5}$	V_{IO}	---
AH28[6]	MD18	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH29[6]	MD48	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH30[6]	MD20	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AH31[6]	MD51	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ1	PLL5B	I/O	IN_T, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST2	O	$O_{2/5}$		PMR[29] = 1
AJ2	X32I	I	WIRE	V_{BAT}	---
AJ3	X32O	O	WIRE	V_{BAT}	---
AJ4	V_{PLL3}	PWR	---	---	---
AJ5[6, 2]	ONCTL#	O	OD_{14}	V_{SB}	---
AJ6	GPWIO2	I/O (PU_{100})	IN_{TS}, $TS_{2/14}$	V_{SB}	---
AJ7	V_{IO}	PWR	---	---	---
AJ8	GPIO11	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	RI2#	I ($PU_{22.5}$)	IN_{TS}		PMR[18] = 1 and PMR[8] = 0
	IRQ15	I ($PU_{22.5}$)	IN_{TS1}		PMR[18] = 0 and PMR[8] = 1
AJ9[6]	MD0	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ10	V_{IO}	PWR	---	---	---
AJ11[6]	MD6	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ12	CASA#	O	$O_{2/5}$	V_{IO}	---
AJ13	BA0	O	$O_{2/5}$	V_{IO}	---
AJ14	MA10	O	$O_{2/5}$	V_{IO}	---
AJ15[6]	MD32	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ16[6]	MD33	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ17[6]	MD36	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ18[6]	MD47	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ19[6]	MD45	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ20[6]	MD42	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ21	SDCLK0	O	$O_{2/5}$	V_{IO}	---
AJ22	V_{IO}	PWR	---	---	---
AJ23	MA6	O	$O_{2/5}$	V_{IO}	---
AJ24	MA3	O	$O_{2/5}$	V_{IO}	---
AJ25	V_{IO}	PWR	---	---	---
AJ26[6]	MD11	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ27	SDCLK_IN	I	IN_T	V_{IO}	---
AJ28[6]	MD19	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ29	V_{IO}	PWR	---	---	---
AJ30[6]	MD22	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AJ31[6]	MD17	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK1	V_{IO}	PWR	---	---	---
AK2	V_{SS}	GND	---	---	---
AK3	AV_{SSPLL3}	GND	---	---	---
AK4	THRM#	I	IN_{TS}	V_{SB}	---
AK5	GPWIO1	I/O (PU_{100})	IN_{TS}, $TS_{2/14}$	V_{SB}	---

Signal Definitions (Continued)

Table 2-4. 481-TEPBGA Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AK6[6, 2]	PWRCNT1	O	OD_{14}	V_{SB}	---
AK7	V_{SS}	GND	---	---	---
AK8	IRRX1	I	IN_{TS}	V_{SB}	PMR[6] = 0
	SIN3	I	IN_{TS}	V_{IO}	PMR[6] =1
AK9[6]	MD1	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK10	V_{SS}	GND	---	---	---
AK11[6]	MD7	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK12	RASA#	O	$O_{2/5}$	V_{IO}	---
AK13	V_{IO}	PWR	---	---	---
AK14	BA1	O	$O_{2/5}$	V_{IO}	---
AK15	MA2	O	$O_{2/5}$	V_{IO}	---
AK16	V_{IO}	PWR		---	---
AK17[6]	MD35	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK18[6]	MD46	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK19	V_{IO}	PWR	---	---	---
AK20[6]	MD43	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK21	DQM5	O	$O_{2/5}$	V_{IO}	---
AK22	V_{SS}	GND	---	---	---
AK23	MA5	O	$O_{2/5}$	V_{IO}	---
AK24[6]	MD15	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK25	V_{SS}	GND	---	---	---
AK26[6]	MD14	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK27[6]	MD12	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK28	SDCLK_OUT	O	$O_{2/5}$	V_{IO}	---
AK29[6]	MD16	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AK30	V_{SS}	GND	---	---	---
AK31	V_{IO}	PWR	---	---	---
AL1	V_{SS}	GND	---	---	---
AL2	V_{IO}	PWR	---	---	---
AL3	V_{BAT}	PWR	---	---	---
AL4	LED#	O	OD_{14}	V_{SB}	---
AL5	V_{SB}	PWR	---	---	---
AL6	V_{SBL}	PWR	---	---	---
AL7[6, 2]	PWRCNT2	O	OD[14]	V_{SB}	---
AL8	SDATA_IN2	I	IN_{TS}	V_{SB}	F3BAR0+Memory Offset 08h[21] = 1
AL9[6]	MD2	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL10[6]	MD4	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL11	DQM0	O	$O_{2/5}$	V_{IO}	---
AL12	CS0#	O	$O_{2/5}$	V_{IO}	---
AL13	V_{SS}	GND	---	---	---
AL14	MA0	O	$O_{2/5}$	V_{IO}	---
AL15	DQM4	O	$O_{2/5}$	V_{IO}	---
AL16	V_{SS}	GND	---	---	---
AL17[6]	MD38	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL18[6]	MD39	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL19	V_{SS}	GND	---	---	---
AL20[6]	MD44	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL21[6]	MD40	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL22	CKEA	O	$O_{2/5}$	V_{IO}	---
AL23	MA7	O	$O_{2/5}$	V_{IO}	---
AL24	MA4	O	$O_{2/5}$	V_{IO}	---
AL25[6]	MD8	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL26[6]	MD10	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL27[6]	MD9	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL28	MA12	O	$O_{2/5}$	V_{IO}	---
AL29[6]	MD23	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
AL30	V_{IO}	PWR	---	---	---
AL31	V_{SS}	GND	---	---	---

1. For Buffer Type definitions, refer to Table 8-9 "Buffer Types" on page 365.
2. Is 5V tolerant (ACK#, AFD#/DSTRB#, BUSY/WAIT#, ERR#, INIT#, PD[7:0], PE, SLCT, SLIN#/ASTRB#, STB#/WRITE#, ONCTL#, PWRCNT[2:1]).
3. The TFT_PRSNT strap determines the power-on reset (POR) state of PMR[23].
4. The LPC_ROM strap determines the power-on reset (POR) state of PMR[14] and PMR[22].
5. May need 5V tolerant protection at system level (DDC_SCL, DDC_SDA).
6. Is back-drive protected (MD[63:0], DPOS_PORT1, DNEG_PORT1, DPOS_PORT2, DNEG_PORT2, DPOS_PORT3, DNEG_PORT3, ACK#, AFD#/DSTRB#, BUSY/WAIT#, ERR#, INIT#, PD[7:0], PE, SLCT, SLIN#/ASTRB#, STB#/WRITE#, ONCTL#, PWRCNT[2:1]).

Signal Definitions (Continued)

Table 2-5. 481-TEPBGA Ball Assignment - Sorted Alphabetically by Signal Name

Signal Name	Ball No.
A0	U1
A1	P3
A2	U3
A3	N1
A4	P1
A5	N3
A6	N2
A7	M2
A8	M4
A9	L2
A10	L3
A11	K1
A12	L4
A13	J1
A14	K4
A15	J3
A16	E1
A17	F4
A18	E3
A19	E2
A20	D3
A21	D1
A22	D2
A23	B6
AB1C	N31
AB1D	N30
AB2C	N29
AB2D	M29
AC97_CLK	P31
AC97_RST#	U29
ACK#	B18
AD0	U1
AD1	P3
AD2	U3
AD3	N1
AD4	P1
AD5	N3
AD6	N2
AD7	M2
AD8	M4
AD9	L2
AD10	L3
AD11	K1
AD12	L4
AD13	J1
AD14	K4
AD15	J3
AD16	E1
AD17	F4
AD18	E3
AD19	E2
AD20	D3
AD21	D1
AD22	D2
AD23	B6
AD24	C2
AD25	C4
AD26	C1
AD27	D4
AD28	B4
AD29	B3
AD30	A3
AD31	D5
AFD#/DSTRB#	D22
AV_{CCCRT}	A12, C13, D15
AV_{CCUSB}	D27
AV_{SSCRT}	B14, C14, C15
AV_{SSPLL2}	C16
AV_{SSPLL3}	AK3
AV_{SSUSB}	C27
BA0	AJ13
BA1	AK14
BHE#	E4
BIT_CLK	U30
BLUE	A15
BOOT16	C8
BUSY/WAIT#	B17
C/BE0#	L1
C/BE1#	J2
C/BE2#	F3
C/BE3#	H4
CASA#	AJ12
CKEA	AL22
CLK27M	AA4
CLK32	AH8
CLKSEL0	B8
CLKSEL1	AF3
CLKSEL2	D29
CLKSEL3	P30
CS0#	AL12
CS1#	AH27
CTS2#	C31
D0	C2
D1	C4
D2	C1
D3	D4
D4	B4
D5	B3
D6	A3
D7	D5
D8	L1
D9	J2
D10	F3
D11	H4
D12	J4
D13	F1
D14	F2
D15	G1
DCD2#	C28
DDC_SCL	Y1
DDC_SDA	Y2
DEVSEL#	E4
DID0	C5
DID1	C6
DNEG_PORT1	A29
DNEG_PORT2	B28
DNEG_PORT3	A27
DOCCS#	A9, N31
DOCR#	D9
DOCW#	A8
DPOS_PORT1	A28
DPOS_PORT2	B27
DPOS_PORT3	A26
DQM0	AL11
DQM1	AH23
DQM2	AG30
DQM3	AC31
DQM4	AL15
DQM5	AK21
DQM6	AG29
DQM7	AB31
DSR2#	B29
DTR1#/BOUT1	AG1
DTR2#/BOUT2	D28
ERR#	D21
F_AD0	C21
F_AD1	A21
F_AD2	D20
F_AD3	C20
F_AD4	C18
F_AD5	C19
F_AD6	A20
F_AD7	A18
F_C/BE0#	D21
F_C/BE1#	B17
F_C/BE2#	D17

Signal Definitions (Continued)

Table 2-5. 481-TEPBGA Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
F_C/BE3#	C17
F_DEVSEL#	V31
F_FRAME#	A22
F_GNT0#	U31
F_IRDY#	B20
F_STOP#	U29
F_TRDY#	U30
FP_VDD_ON	V30, AB1
FPCI_MON	A4
FPCICLK	B18
FRAME#	D8
GNT0#	C5
GNT1#	C6
GPIO0	D11
GPIO1	D10, N30
GPIO6	D28
GPIO7	C30
GPIO8	C31
GPIO9	C28
GPIO10	B29
GPIO11	AJ8
GPIO12	N29
GPIO13	M29
GPIO14	D9
GPIO15	A8
GPIO16	V31
GPIO17	A10
GPIO18	AG1
GPIO19	C9
GPIO20	A9, N31
GPIO32	M28
GPIO33	L31
GPIO34	L30
GPIO35	L29
GPIO36	L28
GPIO37	K31
GPIO38/IRRX2	K28
GPIO39	J31
GPIO40	Y3
GPIO41	W4
GPWIO0	AH6
GPWIO1	AK5
GPWIO2	AJ6
GREEN	A14
GTEST	F30
GXCLK	V30
HSYNC	A11
IDE_ADDR0	AD3
IDE_ADDR1	AE1

Signal Name	Ball No.
IDE_ADDR2	U2
IDE_CS0#	AF2
IDE_CS1#	P2
IDE_DACK0#	AD4
IDE_DACK1#	C30
IDE_DATA0	AC3
IDE_DATA1	AC1
IDE_DATA2	AC2
IDE_DATA3	AB4
IDE_DATA4	AB1
IDE_DATA5	AA4
IDE_DATA6	AA3
IDE_DATA7	AA2
IDE_DATA8	Y3
IDE_DATA9	Y2
IDE_DATA10	Y1
IDE_DATA11	W4
IDE_DATA12	W3
IDE_DATA13	V3
IDE_DATA14	V2
IDE_DATA15	V1
IDE_DREQ0	AC4
IDE_DREQ1	C31
IDE_IOR0#	Y4
IDE_IOR1#	D28
IDE_IORDY0	AD1
IDE_IORDY1	B29
IDE_IOW0#	AD2
IDE_IOW1#	C28
IDE_RST#	AA1
INIT#	B21
INTA#	D26
INTB#	C26
INTC#	C9
INTD#	AA2
INTR_O	D22
IOCHRDY	C9
IOCS0#	A10
IOCS1#	D10
IOCS1#	N30
IOR#	D9
IOW#	A8
IRDY#	F2
IRQ9	AA3
IRQ14	AF1
IRQ15	AJ8
IRRX1	AK8
IRTX	C11
LAD0	M28

Signal Name	Ball No.
LAD1	L31
LAD2	L30
LAD3	L29
LDRQ#	L28
LED#	AL4
LFRAME#	K31
LOCK#	H3
LPC_ROM	D6
LPCPD#	K28
MA0	AL14
MA1	AH15
MA2	AK15
MA3	AJ24
MA4	AL24
MA5	AK23
MA6	AJ23
MA7	AL23
MA8	AH22
MA9	AH21
MA10	AJ14
MA11	AH26
MA12	AL28
MD0	AJ9
MD1	AK9
MD2	AL9
MD3	AH10
MD4	AL10
MD5	AH11
MD6	AJ11
MD7	AK11
MD8	AL25
MD9	AL27
MD10	AL26
MD11	AJ26
MD12	AK27
MD13	AH24
MD14	AK26
MD15	AK24
MD16	AK29
MD17	AJ31
MD18	AH28
MD19	AJ28
MD20	AH30
MD21	AG28
MD22	AJ30
MD23	AL29
MD24	AB28
MD25	AC28
MD26	AC29

Signal Definitions (Continued)

Table 2-5. 481-TEPBGA Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
MD27	AC30
MD28	AE31
MD29	AD29
MD30	AD30
MD31	AD31
MD32	AJ15
MD33	AJ16
MD34	AH16
MD35	AK17
MD36	AJ17
MD37	AH17
MD38	AL17
MD39	AL18
MD40	AL21
MD41	AH20
MD42	AJ20
MD43	AK20
MD44	AL20
MD45	AJ19
MD46	AK18
MD47	AJ18
MD48	AH29
MD49	AF29
MD50	AF28
MD51	AH31
MD52	AD28
MD53	AF31
MD54	AF30
MD55	AG31
MD56	Y31
MD57	W28
MD58	Y28
MD59	Y29
MD60	Y30
MD61	AA29
MD62	AA30
MD63	AA31
NC (Total of 8)	A23, A24, A25, B23, B26, C23, C24, D24
ONCTL#	AJ5
OVER_CUR#	AF4
PAR	J4
PC_BEEP	V31
PCICLK	A7
PCICLK0	A4
PCICLK1	D6
PCIRST#	A6
PD0	C21
PD1	A21

Signal Name	Ball No.
PD2	D20
PD3	C20
PD4	C18
PD5	C19
PD6	A20
PD7	A18
PE	D17
PERR#	H2
PLL2B	AH3
PLL5B	AJ1
PLL6B	AG4
POR#	AH9
POWER_EN	AH1
PWRBTN#	AH5
PWRCNT1	AK6
PWRCNT2	AL7
RASA#	AK12
RD#	B8
RED	B12
REQ0#	B5
REQ1#	A5
RI2#	AJ8
ROMCS#	C8
RTS2#	C30
SDATA_IN	U31
SDATA_IN2	AL8
SDATA_OUT	P29
SDCLK_IN	AJ27
SDCLK_OUT	AK28
SDCLK0	AJ21
SDCLK1	W29
SDCLK2	AA28
SDCLK3	V29
SDTEST0	C30
SDTEST1	B29
SDTEST2	C28
SDTEST3	E28
SDTEST4	C31
SDTEST5	D28
SERIRQ	J31
SERR#	H1
SETRES	B15
SIN1	AG2
SIN2	E28
SIN3	AK8
SLCT	C17
SLIN#/ASTRB#	B20
SMI_O	B21
SOUT1	AF3

Signal Name	Ball No.
SOUT2	D29
SOUT3	C11
STB#/WRITE#	A22
STOP#	G1
SYNC	P30
TCK	E31
TDI	F29
TDN	D31
TDO	E30
TDP	D30
TEST0	AH3
TEST1	AG4
TEST2	AJ1
TEST3	V30
TFT_PRSNT	P29
TFTD0	A9, AD4
TFTD1	A20, AF1
TFTD2	D22, AE1
TFTD3	B17, AD3
TFTD4	D21, U2
TFTD5	B21, AF2
TFTD6	C21, AC3
TFTD7	A21, V1
TFTD8	D20, AC4
TFTD9	C20, AD2
TFTD10	C18, Y4
TFTD11	C19, AD1
TFTD12	D10, AB4
TFTD13	A18, W3
TFTD14	D17, AC2
TFTD15	C17, V3
TFTD16	B20, AC1
TFTD17	A22, V2
TFTDCK	A10, AA1
TFTDE	B18, P2
THRM#	AK4
TMS	F28
TRDE#	D11
TRDY#	F1
TRST#	E29
V_{BAT}	AL3
V_{CCCRT}	D12
V_{CORE} (Total of 28)	N13, N14, N18, N19, P4, P13, P14, P18, P19, P28, T1, T2, T3, T4, T28, T29, T30, T31, U4, U28, V13, V14, V18, V19, W13, W14, W18, W19

Signal Definitions (Continued)

Table 2-5. 481-TEPBGA Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
V_{IO} (Total of 43)	A2, A30, B2, B13, B16, B19, B31, C3, C7, C10, C22, C25, C29, D14, D18, D23, G3, G29, K2, K29, M3, M30, W1, W31, AB3, AB29, AE3, AE29, AH4, AH14, AH18, AJ7, AJ10, AJ22, AJ25, AJ29, AK1, AK13, AK16, AK19, AK31, AL2, AL30
VPCKIN	F31
VPD0	J30
VPD1	J29
VPD2	J28
VPD3	H31
VPD4	H30
VPD5	H29
VPD6	H28
VPD7	G31
V_{PLL2}	A17
V_{PLL3}	AJ4
VREF	D16
V_{SB}	AL5
V_{SBL}	AL6

Signal Name	Ball No.
V_{SS} (Total of 92)	A1, A13, A16, A19, A31, B1, B7, B10, B22, B24, B25, B30, D7, D13, D19, D25, G2, G4, G28, G30, K3, K30, M1, M31, N4, N15, N16, N17, N28, P15, P16, P17, R1, R2, R3, R4, R13, R14, R15, R16, R17, R18, R19, R28, R29, R30, R31, T13, T14, T15, T16, T17, T18, T19, U13, U14, U15, U16, U17, U18, U19, V4, V15, V16, V17, V28, W2, W15, W16, W17, W30, AB2, AB30, AE2, AE4, AE28, AE30, AH7, AH13, AH19, AH25, AK2, AK7, AK10, AK22, AK25, AK30, AL1, AL13, AL16, AL19, AL31
V_{SSCRT}	C12
VSYNC	B11

Signal Name	Ball No.
WEA#	AH12
WR#	B9
X27I	AG3
X27O	AH2
X32I	AJ2
X32O	AJ3

Signal Definitions (Continued)

2.2 STRAP OPTIONS

Several balls are read at power-up that set up the state of the SC2200. These balls are typically multiplexed with other functions that are outputs after the power-up sequence is complete. The SC2200 must read the state of the balls at power-up and the internal PU or PD resistors do not guarantee the correct state will be read. Therefore, it is required that an external PU or PD resistor with a value of 1.5 KΩ be placed on the balls listed in Table 2-6. The value of the resistor is important to ensure that the proper state is read during the power-up sequence. If the ball is not read correctly at power-up, the SC2200 may default to a state that causes it to function improperly, possibly resulting in application failure.

Table 2-6. Strap Options

Strap Option	Muxed With	Ball No.		Nominal Internal PU or PD	External PU/PD Strap Settings		Register References
		EBGA	TEPBGA		Strap = 0 (PD)	Strap = 1 (PU)	
CLKSEL0	RD#	F3	B8	PD_{100}	See Table 3-7 on page 94 for CLKSEL strap options.		GCB+I/O Offset 1Eh[9:8] (aka CCFC register bits [9:8]) (RO): Value programmed at reset by CLKSEL[1:0]. GCB+I/O Offset 10h[3:0] (aka MCCM register bits [3:0]) (RO): Value programmed at reset by CLKSEL[3:0]. GCB+I/O Offset 1Eh[3:0] (aka CCFC register bits [3:0]) (R/W, but write not recommended): Value programmed at reset by CLKSEL[3:0]. **Note:** Values for GCB+I/O Offset 10h[3:0] and 1Eh[3:0] are not the same.
CLKSEL1	SOUT1	B27	AF3	PD_{100}			
CLKSEL2	SOUT2	AK3	D29	PD_{100}			
CLKSEL3	SYNC	AL13	P30	PD_{100}			
BOOT16	ROMCS#	G4	C8	PD_{100}	Enable boot from 8-bit ROM	Enable boot from 16-bit ROM	GCB+I/O Offset 34h[3] (aka MCR register bit 3) (RO): Reads back strap setting. GCB+I/O Offset 34h[14] (R/W): Used to allow the ROMCS# width to be changed under program control.
TFT_PRSNT	SDATA_OUT	AK13	P29	PD_{100}	TFT not muxed onto Parallel Port	TFT muxed onto Parallel Port	GCB+I/O Offset 30h[23] (aka PMR register bit 23) (R/W): Reads back strap setting.
LPC_ROM	PCICLK1	E4	D6	PD_{100}	Disable boot from ROM on LPC bus	Enable boot from ROM on LPC bus	F0BAR1+I/O Offset 10h[15] (R/W): Reads back strap setting and allows LPC ROM to be changed under program control.
FPCI_MON	PCICLK0	D3	A4	PD_{100}	Disable Fast-PCI, INTR_O, and SMI_O monitoring signals.	Enable Fast-PCI, INTR_O, and SMI_O monitoring signals. (Useful during debug.)	GCB+I/O Offset 34h[30] (aka MCR register bit 30) (RO): Reads back strap setting. **Note**: For normal operation, strap this signal low using a 1.5 KΩ resistor.
DID0	GNT0#	D4	C5	PD_{100}	Defines the system-level chip ID.		GCB+I/O Offset 34h[31,29] (aka MCR register bits 31 and 29) (RO): Reads back strap setting. **Note**: GNT0# must have a PU resistor of 1.5 KΩ and GNT1# must have a PD resistor of 1.5 KΩ.
DID1	GNT1#	D2	C6	PD_{100}			

Note: Accuracy of internal PU/PD resistors: 80K to 250K.

Location of the GCB (General Configuration Block) cannot be determined by software. See the *SC2200 Thin Client On a Chip device errata* document.

Signal Definitions (Continued)

2.3 MULTIPLEXING CONFIGURATION

The tables that follow list multiplexing options and their configurations. Certain multiplexing options may be chosen per signal; others are available only for a group of signals.

Where ever a GPIO pin is multiplexed with another function, there is an optional pull-up resistor on this pin; after system reset, the pull-up is present. This pull-up resistor can be disabled by writing Core Logic registers. The configuration is without regard to the selected ball function. The above applies to all pins multiplexed with GPIO, except GPIO12, GPIO13, and GPIO16.

Table 2-7. Two-Signal/Group Multiplexing

EBGA	TEPBGA	Default		Alternate	
		Signal	Configuration	Signal	Configuration
Ball No.		**IDE**		**TFT, CRT, PCI, GPIO, System**	
A26	AD3	IDE_ADDR0	PMR[24] = 0	TFTD3	PMR[24] = 1
C26	AE1	IDE_ADDR1		TFTD2	
C17	U2	IDE_ADDR2		TFTD4	
B24	AC3	IDE_DATA0		TFTD6	
A24	AC1	IDE_DATA1		TFTD16	
D23	AC2	IDE_DATA2		TFTD14	
C23	AB4	IDE_DATA3		TFTD12	
B23	AB1	IDE_DATA4		FP_VDD_ON	
A23	AA4	IDE_DATA5		CLK27M	
C22	AA3	IDE_DATA6		IRQ9	
B22	AA2	IDE_DATA7		INTD#	
A21	Y3	IDE_DATA8		GPIO40	
C20	Y2	IDE_DATA9		DDC_SDA	
A20	Y1	IDE_DATA10		DDC_SCL	
C19	W4	IDE_DATA11		GPIO41	
B19	W3	IDE_DATA12		TFTD13	
A19	V3	IDE_DATA13		TFTD15	
C18	V2	IDE_DATA14		TFTD17	
B18	V1	IDE_DATA15		TFTD7	
C21	Y4	IDE_IOR0#		TFTD10	
A25	AD1	IDE_IORDY0		TFTD11	
C24	AC4	IDE_DREQ0		TFTD8	
D24	AD2	IDE_IOW0#		TFTD9	
A27	AF2	IDE_CS0#		TFTD5	
C16	P2	IDE_CS1#		TFTDE	
C25	AD4	IDE_DACK0#		TFTD0	
A22	AA1	IDE_RST#		TFTDCK	
D25	AF1	IRQ14		TFTD1	
Ball No.		**Sub-ISA**		**GPIO**	
H1	D11	TRDE#	PMR[12] = 0	GPIO0	PMR[12] = 1

Signal Definitions (Continued)

Table 2-7. Two-Signal/Group Multiplexing (Continued)

EBGA	TEPBGA	Default		Alternate	
		Signal	Configuration	Signal	Configuration
Ball No.		**GPIO**		**ACCESS.bus**	
AJ12	N29	GPIO12	PMR[19] = 0	AB2C	PMR[19] = 1
AL11	M29	GPIO13		AB2D	
Ball No.		**GPIO**		**UART**	
A28	AG1	GPIO18	PMR[16] = 0	DTR1#/BOUT1	PMR[16] = 1
Ball No.		**Infrared**		**UART**	
J3	C11	IRTX	PMR[6] = 0	SOUT3	PMR[6] = 1
J28	AK8	IRRX1		SIN3	
Ball No.		**GPIO**		**LPC**	
AJ11	M28	GPIO32	PMR[14] = 0 and PMR[22] = 0	LAD0	PMR[14] = 1 and PMR[22] = 1
AL10	L31	GPIO33		LAD1	
AK10	L30	GPIO34		LAD2	
AJ10	L29	GPIO35		LAD3	
AL9	L28	GPIO36		LDRQ#	
AK9	K31	GPIO37		LFRAME#	
AJ9	K28	GPIO38/IRRX2		LPCPD#	
AL8	J31	GPIO39		SERIRQ	
Ball No.		**UART**		**Internal Test**	
AJ4	E28	SIN2	PMR[28] = 0	SDTEST3	PMR[28] = 1
Ball No.		**AC97**		**FPCI Monitoring**	
AJ15	U29	AC97_RST#	FPCI_MON = 0	F_STOP#	FPCI_MON = 1
AK14	U31	SDATA_IN		F_GNT0#	
AL14	U30	BIT_CLK		F_TRDY#	
Ball No.		**Internal Test**		**Internal Test**	
C28	AG4	PLL6B	PMR[29] = 0	TEST1	PMR[29] = 1
B29	AJ1	PLL5B		TEST2	
D28	AH3	PLL2B		TEST0	

Signal Definitions (Continued)

Table 2-8. Three-Signal/Group Multiplexing

EBGA	TEPBGA	Default		Alternate1		Alternate2	
		Signal	Configuration	Signal	Configuration	Signal	Configuration
Ball No.		**Sub-ISA**		**Sub-ISA[1]**		**GPIO**	
F1	D9	IOR#	PMR[21] = 0 and PMR[2] = 0	DOCR#	PMR[21] = 0 and PMR[2] = 1	GPIO14	PMR[21] = 1 and PMR[2] = 1
G3	A8	IOW#		DOCW#		GPIO15	
Ball No.		**GPIO**		**AC97**		**FPCI Monitoring**	
AL15	V31	GPIO16	PMR[0] = 0 and FPCI_MON = 0	PC_BEEP	PMR[0] = 1 = 0 and FPCI_MON = 0	F_DEVSEL	FPCI_MON = 1
Ball No.		**GPIO**		**PCI[2]**		**Sub-ISA**	
H4	C9	GPIO19	PMR[9] = 0 and PMR[4] = 0	INTC#	PMR[9] = 0 and PMR[4] = 1	IOCHRDY	PMR[9] = 1 and PMR[4] = 1
Ball No.		**Parallel Port**		**TFT[3]**		**FPCI Monitoring**	
U3	B18	ACK#	PMR[23] = 0 and (PMR[27] = 0 and FPCI_MON = 0)	TFTDE	PMR[23] = 1 and (PMR[27] = 0 and FPCI_MON = 0)	FPCI_CLK	PMR[23] = 0 and (PMR[27] = 1 or FPCI_MON = 1)
AB2	D22	AFD#/DSTRB#		TFTD2		INTR_O	
T1	B17	BUSY/WAIT#		TFTD3		F_C/BE1#	
AA3	D21	ERR#		TFTD4		F_C/BE0#	
Y3	B21	INIT#		TFTD5		SMI_O	
AA1	C21	PD0		TFTD6		F_AD0	
Y1	A21	PD1		TFTD7		F_AD1	
W3	D20	PD2		TFTD8		F_AD2	
W2	C20	PD3		TFTD9		F_AD3	
V1	C18	PD4		TFTD10		F_AD4	
V2	C19	PD5		TFTD11		F_AD5	
V3	A20	PD6		TFTD1		F_AD6	
U1	A18	PD7		TFTD13		F_AD7	
T3	D17	PE		TFTD14		F_C/BE2#	
T4	C17	SLCT		TFTD15		F_C/BE3#	
W1	B20	SLIN# /ASTRB#		TFTD16		F_IRDY	
AB1	A22	STB#/WRITE#		TFTD17		F_FRAME#	
Ball No.		**GPIO**		**Sub-ISA**		**TFT[3]**	
J4	A10	GPIO17	PMR[23] = 0 and PMR[5] = 0	IOCS0#	PMR[23] = 0 and PMR[5] = 1	TFTDCK	PMR[23] = 1
H3	A9	GPIO20	PMR[23] = 0 and PMR[7] = 0	DOCCS#	PMR[23] = 0 and PMR[7] = 1	TFTD0	PMR[23] = 1
H2	D10	GPIO1	PMR[23] = 0 and PMR[13] = 0	IOCS1#	PMR[23] = 0 and PMR[13] = 1	TFTD12	PMR[23] = 1
Ball No.		**AB1**		**GPIO**		**Sub-ISA**	
AJ13	N31	AB1C	PMR[23] = 0	GPIO20	PMR[23] = 1 and PMR[7] = 0	DOCCS#	PMR[23] = 1 and PMR[7] = 1
AL12	N30	AB1D	PMR[23] = 0	GPIO1	PMR[23] = 1 and PMR[13] = 0	IOCS1#	PMR[23] = 1 and PMR[13] = 1
Ball No.		**GPIO**		**UART2**		**IDE2**	
H30	AJ8	GPIO11	PMR[18] = 0 and PMR[8] = 0	RI2#	PMR[18] = 1 and PMR[8] = 0	IRQ15	PMR[18] = 0 and PMR[8] = 1

Signal Definitions (Continued)

Table 2-8. Three-Signal/Group Multiplexing (Continued)

EBGA	TEPBGA	Default		Alternate1		Alternate2	
		Signal	Configuration	Signal	Configuration	Signal	Configuration
Ball No.		Internal Test		Internal Test		TFT	
AL16	V30	GXCLK	PMR[23] = 0 and PMR[29] = 0	TEST3	PMR[23] = 0 and PMR[29] = 1	FP_VDD_ON	PMR[23] = 1

1. The combination of PMR[21] = 1 and PMR[2] = 0 is undefined and should not be used.
2. The combination of PMR[9] = 1 and PMR[4] = 0 is undefined and should not be used.
3. These TFT outputs are reset to 0 by POR# if the TFT_PRSNT strap is pulled high or PMR[10] = 0. This relates to signals TFTD[17:0], TFTDE, TFTDCK.

Table 2-9. Four-Signal/Group Multiplexing

EBGA	TEPBGA	Default		Alternate1		Alternate2		Alternate3	
		Signal	Configuration	Signal	Configuration	Signal	Configuration	Signal	Configuration
Ball No.		GPIO		UART2		IDE2		Internal Test	
AH4	C30	GPIO7	PMR[17] = 0 and PMR[8] = 0	RTS2#	PMR[17] = 1 and PMR[8] = 0	IDE_DACK1#	PMR[17] = 0 and PMR[8] = 1	SDTEST0	PMR[17] = 1 and PMR[8] = 1
AJ2	C31	GPIO8		CTS2#		IDE_DREQ1		SDTEST4	
AH3	D28	GPIO6	PMR[18] = 0 and PMR[8] = 0	DTR2#/BOUT2	PMR[18] = 1 and PMR[8] = 0	IDE_IOR1#	PMR[18] = 0 and PMR[8] = 1	SDTEST5	PMR[18] = 1 and PMR[8] = 1
AG4	C28	GPIO9		DCD2#		IDE_IOW1#		SDTEST2	
AJ1	B29	GPIO10		DSR2#		IDE_IORDY1		SDTEST1	

Signal Definitions (Continued)

2.4 SIGNAL DESCRIPTIONS

Information in the tables that follow may have duplicate information in multiple tables. Multiple references all contain identical information.

2.4.1 System Interface

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
CLKSEL1	B27	AF3	I	**Fast-PCI Clock Selects.** These strap signals are used to set the internal Fast-PCI clock. 00 = 33.3 MHz 01 = 48 MHz 10 = 66.7 MHz 11 = 33.3 MHz During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	SOUT1
CLKSEL0	F3	B8			RD#
CLKSEL3	AL13	P30	I	**Maximum Core Clock Multiplier.** These strap signals are used to set the maximum allowed multiplier value for the core clock. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	SYNC
CLKSEL2	AK3	D29			SOUT2
BOOT16	G4	C8	I	**Boot ROM is 16 Bits Wide.** This strap signal enables the optional 16-bit wide Sub-ISA bus. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	ROMCS#
LPC_ROM	E4	D6	I	**LPC_ROM.** This strap signal forces selecting of the LPC bus and sets bit F0BAR1+I/O Offset 10h[15], LPC ROM Addressing Enable. It enables the SC2200 to boot from a ROM connected to the LPC bus. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	PCICLK1
TFT_PRSNT	AK13	P29	I	**TFT Present.** A strap used to select multiplexing of TFT signals at power-up. Enables using TFT instead of Parallel Port, ACB1, and GPIO17. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	SDATA_OUT
FPCI_MON	D3	A4	I	**Fast-PCI Monitoring.** The strap on this ball forces selection of Fast-PCI monitoring signals. For normal operation, strap this signal low using a 1.5 KΩ resistor. The value of this strap can be read on the MCR[30].	PCICLK0

Signal Definitions (Continued)

2.4.1 System Interface (Continued)

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
DID1	D2	C6	I	**Device ID**. Together, the straps on these signals define the system-level chip ID. The value of DID1 can be read in the MCR[29]. The value of DID0 can be read in the MCR[31]. DID0 must have a pull-up resistor of 1.5 KΩ and DID1 must have a pull-down resistor of 1.5 KΩ.	GNT1#
DID0	D4	C5	I		GNT0#
POR#	J29	AH9	I	**Power On Reset.** POR# is the system reset signal generated from the power supply to indicate that the system should be reset.	---
X32I	C30	AJ2	I/O	**Crystal Connections.** Connected directly to a 32.768 KHz crystal. This clock input is required even if the internal RTC is not being used. Some of the internal clocks are derived from this clock. If an external clock is used, it should be connected to X32I, using a voltage level of 0 volts to V_{CORE} +10% maximum. X32O should remain unconnected.	---
X32O	D29	AJ3			---
X27I	A29	AG3	I/O	**Crystal Connections.** Connected directly to a 27.000 MHz crystal. Some of the internal clocks are derived from this clock. If an external clock is used, it should be connected to X27I, using a voltage level of 0 volts to V_{IO} and X27O should be remain unconnected.	---
X27O	D27	AH2			---
CLK27M	A23	AA4	O	**27 MHz Output Clock.** Output of crystal oscillator.	IDE_DATA5
PCIRST#	D1	A6	O	**PCI and System Reset.** PCIRST# is the reset signal for the PCI bus and system. It is asserted for approximately 100 µs after POR# is negated.	---

2.4.2 Memory Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
MD[63:0]	See Table 2-3 on page 32.	See Table 2-5 on page 47.	I/O	**Memory Data Bus.** The data bus lines driven to/from system memory.	---
MA[12:0]	See Table 2-3 on page 32.	See Table 2-5 on page 47.	O	**Memory Address Bus.** The multiplexed row/column address lines driven to the system memory. Supports 256-Mbit SDRAM.	---
BA1	P31	AK14	O	**Bank Address Bits.** These bits are used to select the component bank within the SDRAM.	---
BA0	P30	AJ13			---

Signal Definitions (Continued)

2.4.2 Memory Interface Signals (Continued)

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
CS1#	AK29	AH27	O	**Chip Selects.** These bits are used to select the module bank within system memory. Each chip select corresponds to a specific module bank. If CS# is high, the bank(s) do not respond to RAS#, CAS#, and WE# until the bank is selected again.	---
CS0#	P29	AL12			---
RASA#	N31	AK12	O	**Row Address Strobe.** RAS#, CAS#, WE# and CKE are encoded to support the different SDRAM commands. RASA# is used with CS[1:0]#.	---
CASA#	N30	AJ12	O	**Column Address Strobe.** RAS#, CAS#, WE# and CKE are encoded to support the different SDRAM commands. CASA# is used with CS[1:0]#.	---
WEA#	N29	AH12	O	**Write Enable.** RAS#, CAS#, WE# and CKE are encoded to support the different SDRAM commands. WEA# is used with CS[1:0]#.	---
DQM7	AJ20	AB31	O	**Data Mask Control Bits.** During memory read cycles, these outputs control whether SDRAM output buffers are driven on the MD bus or not. All DQM signals are asserted during read cycles. During memory write cycles, these outputs control whether or not MD data is written into SDRAM. DQM[7:0] connect directly to the [DQM7:0] pins of each DIMM connector.	---
DQM6	AJ26	AG29			---
DQM5	AC30	AK21			---
DQM4	T28	AL15			---
DQM3	AJ21	AC31			---
DQM2	AL26	AG30			---
DQM1	AF31	AH23			---
DQM0	M31	AL11			---
CKEA	AC28	AL22	O	**Clock Enable.** These signals are used to enter Suspend/power-down mode. CKEA is used with CS[1:0]#. If CKE goes low when no read or write cycle is in progress, the SDRAM enters power-down mode. To ensure that SDRAM data remains valid, the self-refresh command is executed. To exit this mode, and return to normal operation, drive CKE high. These signals should have an external pull-down resistor of 33 KΩ	---
SDCLK3	AJ16	V29	O	**SDRAM Clocks.** SDRAM uses these clocks to sample all control, address, and data lines. To ensure that the Suspend mode functions correctly, SDCLK3 and SDCLK1 should be used with CS1#. SDCLK2 and SDCLK0 should be used together with CS0#.	---
SDCLK2	AL20	AA28			---
SDCLK1	AH16	W29			---
SDCLK0	AC29	AJ21			---
SDCLK_IN	AJ30	AJ27	I	**SDRAM Clock Input.** The SC2200 samples the memory read data on this clock. Works in conjunction with the SDCLK_OUT signal.	---

Signal Definitions (Continued)

2.4.2 Memory Interface Signals (Continued)

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
SDCLK_OUT	AH28	AK28	O	**SDRAM Clock Output.** This output is routed back to SDCLK_IN. The board designer should vary the length of the board trace to control skew between SDCLK_IN and SDCLK.	---

2.4.3 Video Port Interface Signals

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
VPD7	AJ6	G31	I	**Video Port Data.** The data is input from the CCIR-656 video decoder.	---
VPD6	AJ7	H28			---
VPD5	AL6	H29			---
VPD4	AH8	H30			---
VPD3	AL7	H31			---
VPD2	AJ8	J28			---
VPD1	AK8	J29			---
VPD0	AH9	J30			---
VPCKIN	AH7	F31	I	**Video Port Clock Input.** The clock input from the video decoder.	---

Signal Definitions (Continued)

2.4.4 CRT/TFT Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
DDC_SCL	A20	Y1	O	**DDC Serial Clock.** This is the serial clock for the VESA Display Data Channel interface. It is used for monitor communications. The DDC2B standard is supported by this interface.	IDE_DATA10
DDC_SDA	C20	Y2	I/O	**DDC Serial Data.** This is the bidirectional serial data signal for the VESA Display Data Channel interface. It is used for monitor communications. The DDC2B standard is supported by this interface.	IDE_DATA9
HSYNC	J1	A11	O	**Horizontal Sync**	---
VSYNC	J2	B11	O	**Vertical Sync**	---
VREF	P1	D16	I/O	**Voltage Reference.** Reference voltage for CRT PLL and DAC. This signal reflects the internal voltage reference. If internal voltage reference is used (recommended), leave this ball disconnected. If an external voltage reference is used, this input is tied to a 1.235V reference.	---
SETRES	P2	B15	I	**Set Resistor.** This signal sets the current level for the RED/GREEN/BLUE analog outputs. Typically, a 464Ω, 1% resistor is connected between this ball and AV_{SSCRT}.	---
On-Chip RAMDAC					
RED	K1	B12	O	**Analog Red, Green and Blue**	---
GREEN	M3	A14			---
BLUE	N2	A15			---
TFT (External DAC) Interface					
TFTDCK	A22	AA1	O	**TFT Clock.** Clock to external CRT DACs or TFT.	IDE_RST#
	J4	A10			GPIO17+ IOCS0#
TFTDE	C16	P2	O	**TFT Data Enable.** Can be used as blank signal to external CRT DACs.	IDE_CS1#
	U3	B18			ACK#+FPCICLK
FP_VDD_ON	B23	AB1	O	**TFT Power Control.** Used to enable power to the Flat Panel display, with power sequence timing.	IDE_DATA4
	AL16	V30			GXCLK+TEST3
TFTD[17:0]	See Table 2-3 on page 32.	See Table 2-5 on page 47.	O	**Digital RGB Data to TFT.** TFTD[5:0] - Connect to BLUE TFT inputs. TFTD[11:6] - Connect to GREEN TFT inputs. TFTD[17:12] - Connect to RED TFT inputs.	The TFT interface is muxed with the IDE interface or the Parallel Port. See Table 2-7 on page 52 and Table 2-8 on page 54 for details.

Signal Definitions (Continued)

2.4.5 ACCESS.bus Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
AB1C	AJ13	N31	I/O	**ACCESS.bus 1 Serial Clock.** This is the serial clock for the interface. **Note:** If selected as AB1C function but not used, tie AB1C high.	GPIO20+DOCCS#
AB1D	AL12	N30	I/O	**ACCESS.bus 1 Serial Data.** This is the bidirectional serial data signal for the interface. **Note:** If AB1D function is selected but not used, tie AB1D high.	GPIO1+IOCS1#
AB2C	AJ12	N29	I/O	**ACCESS.bus 2 Serial Clock.** This is the serial clock for the interface. **Note:** If AB2C function is selected but not used, tie AB2C high.	GPIO12
AB2D	AL11	M29	I/O	**ACCESS.bus 2 Serial Data.** This is the bidirectional serial data signal for the interface. **Note:** If AB2D function is selected but not used, tie AB2D high.	GPIO13

2.4.6 PCI Bus Interface Signals

Signal Name	BalL No. EBGA	BalL No. TEPBGA	Type	Description	Mux
PCICLK	E2	A7	I	**PCI Clock.** PCICLK provides timing for all transactions on the PCI bus. All other PCI signals are sampled on the rising edge of PCICLK, and all timing parameters are defined with respect to this edge.	---
PCICLK0	D3	A4	O	**PCI Clock Outputs.** PCICLK0 and PCICLK1 provide clock drives for the system at 33 MHz. These clocks are asynchronous to PCI signals. There is low skew between all outputs. One of these clock signals should be connected to the PCICLK input. All PCI clock users in the system (including PCICLK) should receive the clock with as low a skew as possible.	FPCI_MON (Strap)
PCICLK1	E4	D6	O		LPC_ROM (Strap)
AD[31:24]	See Table 2-3 on page 32.	See Table 2-5 on page 47.	I/O	**Multiplexed Address and Data.** A bus transaction consists of an address phase in the cycle in which FRAME# is asserted followed by one or more data phases. During the address phase, AD[31:0] contain a physical 32-bit address. For I/O, this is a byte address. For configuration and memory, it is a DWORD address. During data phases, AD[7:0] contain the least significant byte (LSB) and AD[31:24] contain the most significant byte (MSB).	D[7:0]
AD[23:0]					A[23:0]

Signal Definitions (Continued)

2.4.6 PCI Bus Interface Signals (Continued)

Signal Name	BalL No. EBGA	BalL No. TEPBGA	Type	Description	Mux
C/BE3#	A8	H4	I/O	**Multiplexed Command and Byte Enables.** During the address phase of a transaction when FRAME# is active, C/BE[3:0]# define the bus command. During the data phase, C/BE[3:0]# are used as byte enables. The byte enables are valid for the entire data phase and determine which byte lanes carry meaningful data. C/BE0# applies to byte 0 (LSB) and C/BE3# applies to byte 3 (MSB).	D11
C/BE2#	D8	F3			D10
C/BE1#	A10	J2			D9
C/BE0#	A13	L1			D8
INTA#	AE3	D26	I	**PCI Interrupts.** The SC2200 provides inputs for the optional “level-sensitive” PCI interrupts (also known in industry terms as PIRQx#). These interrupts can be mapped to IRQs of the internal 8259A interrupt controllers using PCI Interrupt Steering Registers 1 and 2 (F0 Index 5Ch and 5Dh). **Note:** If selected as INTC# or INTD# function(s) but not used, tie INTC# and INTD# high.	---
INTB#	AF1	C26			---
INTC#	H4	C9			GPIO19+IOCHRDY
INTD#	B22	AA2			IDE_DATA7
PAR	C10	J4	I/O	**Parity.** Parity generation is required by all PCI agents. The master drives PAR for address- and write-data phases. The target drives PAR for read-data phases. Parity is even across AD[31:0] and C/BE[3:0]#. For address phases, PAR is stable and valid one PCI clock after the address phase. It has the same timing as AD[31:0] but is delayed by one PCI clock. For data phases, PAR is stable and valid one PCI clock after either IRDY# is asserted on a write transaction or after TRDY# is asserted on a read transaction. Once PAR is valid, it remains valid until one PCI clock after the completion of the data phase. (Also see PERR#.)	D12
FRAME#	E1	D8	I/O	**Frame Cycle.** Frame is driven by the current master to indicate the beginning and duration of an access. FRAME# is asserted to indicate the beginning of a bus transaction. While FRAME# is asserted, data transfers continue. FRAME# is de-asserted when the transaction is in the final data phase. This signal is internally connected to a pull-up resistor.	---

Signal Definitions (Continued)

2.4.6 PCI Bus Interface Signals (Continued)

Signal Name	BalL No. EBGA	BalL No. TEPBGA	Type	Description	Mux
IRDY#	C8	F2	I/O	**Initiator Ready.** IRDY# is asserted to indicate that the bus master is able to complete the current data phase of the transaction. IRDY# is used in conjunction with TRDY#. A data phase is completed on any PCI clock in which both IRDY# and TRDY# are sampled as asserted. During a write, IRDY# indicates that valid data is present on AD[31:0]. During a read, it indicates that the master is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together. This signal is internally connected to a pull-up resistor.	D14
TRDY#	B8	F1	I/O	**Target Ready.** TRDY# is asserted to indicate that the target agent is able to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is complete on any PCI clock in which both TRDY# and IRDY# are sampled as asserted. During a read, TRDY# indicates that valid data is present on AD[31:0]. During a write, it indicates that the target is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together. This signal is internally connected to a pull-up resistor.	D13
STOP#	D9	G1	I/O	**Target Stop.** STOP# is asserted to indicate that the current target is requesting that the master stop the current transaction. This signal is used with DEVSEL# to indicate retry, disconnect, or target abort. If STOP# is sampled active by the master, FRAME# is deasserted and the cycle is stopped within three PCI clock cycles. As an input, STOP# can be asserted in the following cases: 1) If a PCI master tries to access memory that has been locked by another master. This condition is detected if FRAME# and LOCK# are asserted during an address phase. 2) If the PCI write buffers are full or if a previously buffered cycle has not completed. 3) On read cycles that cross cache line boundaries. This is conditional based upon the programming of GX1 module's PCI Configuration Register, Index 41h[1]. This signal is internally connected to a pull-up resistor.	D15

Signal Definitions (Continued)

2.4.6 PCI Bus Interface Signals (Continued)

Signal Name	BalL No. EBGA	BalL No. TEPBGA	Type	Description	Mux
LOCK#	C9	H3	I/O	**Lock Operation.** LOCK# indicates an atomic operation that may require multiple transactions to complete. When LOCK# is asserted, non-exclusive transactions may proceed to an address that is not currently locked (at least 16 bytes must be locked). A grant to start a transaction on PCI does not guarantee control of LOCK#. Control of LOCK# is obtained under its own protocol in conjunction with GNT#. It is possible for different agents to use PCI while a single master retains ownership of LOCK#. The arbiter can implement a complete system lock. In this mode, if LOCK# is active, no other master can gain access to the system until the LOCK# is de-asserted. This signal is internally connected to a pull-up resistor.	---
DEVSEL#	B5	E4	I/O	**Device Select.** DEVSEL# indicates that the driving device has decoded its address as the target of the current access. As an input, DEVSEL# indicates whether any device on the bus has been selected. DEVSEL# is also driven by any agent that has the ability to accept cycles on a subtractive decode basis. As a master, if no DEVSEL# is detected within and up to the subtractive decode clock, a master abort cycle is initiated (except for special cycles which do not expect a DEVSEL# returned). This signal is internally connected to a pull-up resistor.	BHE#
PERR#	B9	H2	I/O	**Parity Error.** PERR# is used for reporting data parity errors during all PCI transactions except a Special Cycle. The PERR# line is driven two PCI clocks after the data in which the error was detected. This is one PCI clock after the PAR that is attached to the data. The minimum duration of PERR# is one PCI clock for each data phase in which a data parity error is detected. PERR# must be driven high for one PCI clock before being placed in TRI-STATE. A target asserts PERR# on write cycles if it has claimed the cycle with DEVSEL#. The master asserts PERR# on read cycles. This signal is internally connected to a pull-up resistor.	---

Signal Definitions (Continued)

2.4.6 PCI Bus Interface Signals (Continued)

Signal Name	BalL No. EBGA	BalL No. TEPBGA	Type	Description	Mux
SERR#	A9	H1	I/O	**System Error.** SERR# can be asserted by any agent for reporting errors other than PCI parity, so that the PCI central agent notifies the processor. When the Parity Enable bit is set in the Memory Controller Configuration register, SERR# is asserted upon detection of a parity error in read operations from DRAM. This signal is internally connected to a pull-up resistor.	---
REQ1#	E3	A5	I	**Request Lines.** REQ[1:0]# indicate to the arbiter that an agent requires the bus. Each master has its own REQ# line. REQ# priorities (in order) are: 1) VIP 2) IDE Channel 0 3) IDE Channel 1 4) Audio 5) USB 6) External REQ0# 7) External REQ1# Each REQ# is internally connected to a pull-up resistor.	---
REQ0#	C1	B5			---
GNT1#	D2	C6	O	**Grant Lines.** GNT[1:0]# indicate to the requesting master that it has been granted access to the bus. Each master has its own GNT# line. GNT# can be retracted at any time a higher REQ# is received or if the master does not begin a cycle within a minimum period of time (16 PCI clocks). Each of these signals is internally connected to a pull-up resistor. GNT0# must have a pull-up resistor of 1.5 KΩ and GNT1# must have a pull-down resistor of 1.5 KΩ.	DID1 (Strap)
GNT0#	D4	C5			DID0 (Strap)

Signal Definitions (Continued)

2.4.7 Sub-ISA Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
A[23:0]	See Table 2-3 on page 32.	See Table 2-5 on page 47.	O	**Address Lines**	AD[23:0]
D15	See Table 2-3 on page 32.	See Table 2-5 on page 47.	I/O	**Data Bus**	STOP#
D14					IRDY#
D13					TRDY#
D12					PAR
D11					C/BE3#
D10					C/BE2#
D9					C/BE1#
D8					C/BE0#
D[7:0]					AD[31:24]
BHE#	B5	E4	O	**Byte High Enable.** With A0, defines byte accessed for 16 bit wide bus cycles.	DEVSEL#
IOCS1#	H2	D10	O	**I/O Chip Selects**	GPIO1+TFTD12
	AL12	N30			AB1D+GPIO1
IOCS0#	J4	A10			GPIO17+TFTDCK
ROMCS#	G4	C8	O	**ROM or Flash ROM Chip Select**	BOOT16 (Strap)
DOCCS#	H3	A9	O	**DiskOnChip or NAND Flash Chip Select**	GPIO20+TFTD0
	AJ13	N31			AB1C+GPIO20
TRDE#	H1	D11	O	**Transceiver Data Enable Control.** Active low for Sub-ISA data transfers. The signal timing is as follows: • In a read cycle, TRDE# has the same timing as RD#. • In a write cycle, TRDE# is asserted (to active low) at the time WR# is asserted. It continues being asserted for one PCI clock cycle after WR# has been negated, then it is negated.	GPIO0
RD#	F3	B8	O	**Memory or I/O Read.** Active on any read cycle.	CLKSEL0 (Strap)
WR#	G1	B9	O	**Memory or I/O Write.** Active on any write cycle.	---
IOR#	F1	D9	O	**I/O Read.** Active on any I/O read cycle.	DOCR#+GPIO14
IOW#	G3	A8	O	**I/O Write.** Active on any I/O write cycle.	DOCW#+GPIO15
DOCR#	F1	D9	O	**DiskOnChip or NAND Flash Read.** Active on any memory read cycle to DiskOnChip.	IOR#+GPIO14
DOCW#	G3	A8	O	**DiskOnChip or NAND Flash Write.** Active on any memory write cycle to DiskOnChip.	IOW#+GPIO15

Signal Definitions (Continued)

2.4.7 Sub-ISA Interface Signals (Continued)

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
IRQ9	C22	AA3	I	**Interrupt 9 Request Input.** Active high. **Note:** If IRQ9 function is selected but not used, tie IRQ9 low.	IDE_DATA6
IOCHRDY	H4	C9	I	**I/O Channel Ready** **Note:** If IOCHRDY function is selected but not used, tie IOCHRDY high.	GPIO19+INTC#

2.4.8 Low Pin Count (LPC) Bus Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
LAD3	AJ10	L29	I/O	**LPC Address-Data.** Multiplexed command, address, bidirectional data, and cycle status.	GPIO35
LAD2	AK10	L30			GPIO34
LAD1	AL10	L31			GPIO33
LAD0	AJ11	M28			GPIO32
LDRQ#	AL9	L28	I	**LPC DMA Request.** Encoded DMA request for LPC interface. **Note:** If LDRQ# function is selected but not used, tie LDRQ# high.	GPIO36
LFRAME#	AK9	K31	O	**LPC Frame.** A low pulse indicates the beginning of a new LPC cycle or termination of a broken cycle.	GPIO37
LPCPD#	AJ9	K28	O	**LPC Power-Down.** Signals the LPC device to prepare for power shut-down on the LPC interface.	GPIO38/IRRX2
SERIRQ	AL8	J31	I/O	**Serial IRQ.** The interrupt requests are serialized over a single signal, where each IRQ level is delivered during a designated time slot. **Note:** If SERIRQ function is selected but not used, tie SERIRQ high.	GPIO39

Signal Definitions (Continued)

2.4.9 IDE Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
IDE_RST#	A22	AA1	O	**IDE Reset.** This signal resets all the devices that are attached to the IDE interface.	TFTDCK
IDE_ADDR2	C17	U2	O	**IDE Address Bits.** These address bits are used to access a register or data port in a device on the IDE bus.	TFTD4
IDE_ADDR1	C26	AE1			TFTD2
IDE_ADDR0	A26	AD3			TFTD3
IDE_DATA[15:0]	See Table 2-3 on page 32.	See Table 2-5 on page 47.	I/O	**IDE Data Lines.** IDE_DATA[15:0] transfers data to/from the IDE devices.	The IDE interface is muxed with the TFT interface. See Table 2-7 on page 52 for details.
IDE_IOR0#	C21	Y4	O	**IDE I/O Read Channels 0 and 1.** IDE_IOR0# is the read signal for Channel 0 and IDE_IOR1# is the read signal for Channel 1. Each signal is asserted at read accesses to the corresponding IDE port addresses.	TFTD10
IDE_IOR1#	AH3	D28	O		GPIO6+DTR2#/ BOUT2+SDTEST5
IDE_IOW0#	D24	AD2	O	**IDE I/O Write Channels 0 and 1.** IDE_IOW0# is the write signal for Channel 0. IDE_IOW1# is the write signal for Channel 1. Each signal is asserted at write accesses to corresponding IDE port addresses.	TFTD9
IDE_IOW1#	AG4	C28	O		GPIO9+DCD2#+ SDTEST2
IDE_CS0#	A27	AF2	O	**IDE Chip Selects 0 and 1.** These signals are used to select the command block registers in an IDE device.	TFTD5
IDE_CS1#	C16	P2	O		TFTDE
IDE_IORDY0	A25	AD1	I	**I/O Ready Channels 0 and 1.** When de-asserted, these signals extend the transfer cycle of any host register access if the required device is not ready to respond to the data transfer request. **Note:** If selected as IDE_IORDY0 or IDE_IORDY1 function(s) but not used, then signal(s) should be tied high.	TFTD11
IDE_IORDY1	AJ1	B29	I		GPIO10+DSR2#+ SDTEST1
IDE_DREQ0	C24	AC4	I	**DMA Request Channels 0 and 1.** The IDE_DREQ signals are used to request a DMA transfer from the SC2200. The direction of transfer is determined by the IDE_IOR/IOW signals. **Note:** If selected as IDE_DREQ0/ IDE_DREQ1 function but not used, tie IDE_DREQ0/IDE_DREQ1 low.	TFTD8
IDE_DREQ1	AJ2	C31	I		GPIO8+CTS2# +SDTEST4
IDE_DACK0#	C25	AD4	O	**DMA Acknowledge Channels 0 and 1.** The IDE_DACK# signals acknowledge the DREQ request to initiate DMA transfers.	TFTD0
IDE_DACK1#	AH4	C30	O		GPIO7+RTS2# +SDTEST0

Signal Definitions (Continued)

2.4.9 IDE Interface Signals (Continued)

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
IRQ14	D25	AF1	I	**Interrupt Request Channels 0 and 1.** These input signals are edge-sensitive interrupts that indicate when the IDE device is requesting a CPU interrupt service. **Note:** If selected as IRQ14/IRQ15 function but not used, tie IRQ14/IRQ15 low.	TFTD1
IRQ15	H30	AJ8	I		GPIO11+RI2#

2.4.10 Universal Serial Bus (USB) Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
POWER_EN	B28	AH1	O	**Power Enable.** This signal enables the power to a self-powered USB hub.	---
OVER_CUR#	C27	AF4	I	**Overcurrent.** This signal indicates that the USB hub has detected an overcurrent on the USB.	---
DPOS_PORT1	AH2	A28	I/O	**USB Port 1 Data Positive for Port 1.**	---
DNEG_PORT1	AG3	A29	I/O	**USB Port 1 Data Negative for port 1.**	---
DPOS_PORT2	AH1	B27	I/O	**USB Port 2 Data Positive for Port 2.**	---
DNEG_PORT2	AG2	B28	I/O	**USB Port 2 Data Negative for Port 2.**	---
DPOS_PORT3	AE4	A26	I/O	**USB Port 3 Data Positive for Port 3.**	---
DNEG_PORT3	AF3	A27	I/O	**USB Port 3 Data Negative for Port 3.**	---

Signal Definitions (Continued)

2.4.11 Serial Ports (UARTs) Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
SIN1	D26	AG2	I	**Serial Inputs.** Receive composite serial data from the communications link (peripheral device, modem or other data transfer device). **Note:** If selected as SIN2 or SIN3 function(s) but not used, then signal(s) should be tied high.	---
SIN2	AJ4	E28			SDTEST3
SIN3	J28	AK8			IRRX1
SOUT1	B27	AF3	O	**Serial Outputs.** Send composite serial data to the communications link (peripheral device, modem or other data transfer device). These signals are set active high after a system reset.	CLKSEL1 (Strap)
SOUT2	AK3	D29			CLKSEL2 (Strap)
SOUT3	J3	C11			IRTX
RTS2#	AH4	C30	O	**Request to Send.** When low, indicates to the modem or other data transfer device that the corresponding UART is ready to exchange data. A system reset sets these signals to inactive high, and loopback operation holds them inactive.	GPIO7+ IDE_DACK1#
CTS2#	AJ2	C31	I	**Clear to Send.** When low, indicates that the modem or other data transfer device is ready to exchange data. **Note:** If selected as CTS2# function but not used, tie CTS2# low.	GPIO8+ IDE_DREQ1
DTR1#/BOUT1	A28	AG1	O	**Data Terminal Ready Outputs.** When low, indicate to the modem or other data transfer device that the UART is ready to establish a communications link. After a system reset, these balls provide the DTR# function and set these signals to inactive high. Loopback operation drive them inactive. **Baud Outputs.** Provide the associated serial channel baud rate generator output signal if test mode is selected (i.e., bit 7 of the EXCR1 Register is set).	GPIO18
DTR2#/BOUT2	AH3	D28			GPIO6+IDE_IOR1#
RI2#	H30	AJ8	I	**Ring Indicator.** When low, indicates to the modem that a telephone ring signal has been received by the modem. They are monitored during power-off for wakeup event detection. **Note:** If selected as RI2# function but not used, tie RI2# high.	GPIO11+IRQ15
DCD2#	AG4	C28	I	**Data Carrier Detected.** When low, indicates that the data transfer device (e.g., modem) is ready to establish a communications link. **Note:** If selected as DCD2# function but not used, tie DCD2# high.	GPIO9+IDE_IOW1# +SDTEST2
DSR2#	AJ1	B29	I	**Data Set Ready.** When low, indicates that the data transfer device (e.g., modem) is ready to establish a communications link. **Note:** If selected as DSR2# function but not used, tie DSR2# low.	GPIO10+ IDE_IORDY1

Signal Definitions (Continued)

2.4.12 Parallel Port Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
ACK#	U3	B18	I	**Acknowledge.** Pulsed low by the printer to indicate that it has received data from the Parallel Port.	TFTDE+FPCICLK
AFD#/DSTRB#	AB2	D22	O	**Automatic Feed.** When low, instructs the printer to automatically feed a line after printing each line. This signal is in TRI-STATE after a 0 is loaded into the corresponding control register bit. An external 4.7 KΩ pull-up resistor should be attached to this ball. **Data Strobe (EPP).** Active low, used in EPP mode to denote a data cycle. When the cycle is aborted, DSTRB# becomes inactive (high).	TFTD2+INTR_O
BUSY/WAIT#	T1	B17	I	**Busy.** Set high by the printer when it cannot accept another character. **Wait.** In EPP mode, the Parallel Port device uses this active low signal to extend its access cycle.	TFTD3+F_C/BE1#
ERR#	AA3	D21	I	**Error.** Set active low by the printer when it detects an error.	TFTD4+F_C/BE0#
INIT#	Y3	B21	O	**Initialize.** When low, initializes the printer. This signal is in TRI-STATE after a 1 is loaded into the corresponding control register bit. Use an external 4.7 KΩ pull-up resistor.	TFTD5+SMI_O
PD7	U1	A18	I/O	**Parallel Port Data.** Transfer data to and from the peripheral data bus and the appropriate Parallel Port data register. These signals have a high current drive capability.	TFTD13+F_AD7
PD6	V3	A20			TFTD1+F_AD6
PD5	V2	C19			TFTD11+F_AD5
PD4	V1	C18			TFTD10+F_AD4
PD3	W2	C20			TFTD9+F_AD3
PD2	W3	D20			TFTD8+F_AD2
PD1	Y1	A21			TFTD7+F_AD1
PD0	AA1	C21			TFTD6+F_AD0
PE	T3	D17	I	**Paper End.** Set high by the printer when it is out of paper. This ball has an internal weak pull-up or pull-down resistor that is programmed by software.	TFTD14+F_C/BE2#
SLCT	T4	C17	I	**Select.** Set active high by the printer when the printer is selected.	TFTD15+F_C/BE3#
SLIN#/ASTRB#	W1	B20	O	**Select Input.** When low, selects the printer. This signal is in TRI-STATE after a 0 is loaded into the corresponding control register bit. Uses an external 4.7 KΩ pull-up resistor. **Address Strobe (EPP).** Active low, used in EPP mode to denote an address or data cycle. When the cycle is aborted, ASTRB# becomes inactive (high).	TFTD16+ F_IRDY#

Signal Definitions (Continued)

2.4.12 Parallel Port Interface Signals (Continued)

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
STB#/WRITE#	AB1	A22	O	**Data Strobe.** When low, indicates to the printer that valid data is available at the printer port. This signal is in TRI-STATE after a 0 is loaded into the corresponding control register bit. An external 4.7 KΩ pull-up resistor should be employed. **Write Strobe.** Active low, used in EPP mode to denote an address or data cycle. When the cycle is aborted, WRITE# becomes inactive (high).	TFTD17+ F_FRAME#

2.4.13 Fast Infrared (IR) Port Interface Signals

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
IRRX1	J28	AK8	I	**IR Receive**. Primary input to receive serial data from the IR transceiver. Monitored during power-off for wakeup event detection. **Note:** If selected as IRRX1 function but not used, tie IRRX1 high.	SIN3
IRRX2/GPIO38	AJ9	K28	I	**IR Receive 2**. Auxiliary IR receiver input to support a second transceiver. This input signal can be used when GPIO38 is selected using PMR[14], and when AUX_IRRX bit in register IRCR2 of the IR module in internal SuperI/O is set.	LPCPD#
IRTX	J3	C11	O	**IR Transmit**. IR serial output data.	SOUT3

Signal Definitions (Continued)

2.4.14 AC97 Audio Interface Signals

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
BIT_CLK	AL14	U30	I	**Audio Bit Clock.** The serial bit clock from the codec. **Note:** If selected as BIT_CLK function but not used, tie BIT_CLK low.	F_TRDY#
SDATA_OUT	AK13	P29	O	**Serial Data Output.** This output transmits audio serial data to the codec.	TFT_PRSNT (Strap)
SDATA_IN	AK14	U31	I	**Serial Data Input.** This input receives serial data from the primary codec. **Note:** If selected as SDATA_IN function but not used, tie SDATA_IN low.	F_GNT0#
SDATA_IN2	H31	AL8	I	**Serial Data Input 2.** This input receives serial data from the secondary codec. This signal has wakeup capability.	---
SYNC	AL13	P30	O	**Serial Bus Synchronization.** This bit is asserted to synchronize the transfer of data between the SC2200 and the AC97 codec.	CLKSEL3 (Strap)
AC97_CLK	AJ14	P31	O	**Codec Clock.** It is twice the frequency of the Audio Bit Clock.	---
AC97_RST#	AJ15	U29	O	**Codec Reset.** S3 to S5 wakeup is not supported because AC97_RST# is powered by V_{IO}. If wakeup from states S3 to S5 are needed, a circuit in the system board should be used to reset the AC97 codec.	F_STOP#
PC_BEEP	AL15	V31	O	**PC Beep.** Legacy PC/AT speaker output.	GPIO16+ F_DEVSEL#

Signal Definitions (Continued)

2.4.15 Power Management Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
CLK32	H29	AH8	O	**32.768 KHz Output Clock**	---
GPWIO0	E31	AH6	I/O	**General Purpose Wakeup I/Os.** These signals each have an internal pull-up of 100 KΩ.	---
GPWIO1	G28	AK5			---
GPWIO2	G29	AJ6			---
LED#	D31	AL4	O	**LED Control.** Drives an externally connected LED (on, off or a 1 Hz blink). Sleeping / Working indicator. This signal is an open-drain output.	---
ONCTL#	E30	AJ5	O	**On / Off Control.** This signal indicates to the main power supply that power should be turned on. This signal is an open-drain output.	---
PWRBTN#	E29	AH5	I	**Power Button.** Input used by the power management logic to monitor external system events, most typically a system on/off button or switch. The signal has an internal pull-up of 100 KΩ, a Schmitt-trigger input buffer and debounce protection of at least 16 ms. ACPI is non-functional when the power-up sequence does not include using the power button. If ACPI functionality is desired, the power button must be toggled. This can be done externally or internally. GPIO63 is internally connected to PWRBTN#. To toggle the power button with software, GPIO63 must be programmed as an output using the normal GPIO programming protocol (see Section 5.4.1.1 "GPIO Support Registers" on page 231). GPIO63 must be pulsed low for at least 16 ms and not more than 4 sec. Asserting POR# has no effect on ACPI. If POR# is asserted and ACPI was active prior to POR#, then ACPI will remain active after POR#. Therefore, BIOS must ensure that ACPI is inactive before GPIO63 is pulsed low.	---
PWRCNT1	F31	AK6	O	**Suspend Power Plane Control 1 and 2.** Control signal asserted during power management Suspend states. These signals are open-drain outputs.	---
PWRCNT2	G31	AL7	O		---
THRM#	F28	AK4	I	**Thermal Event.** Active low signal generated by external hardware indicating that the system temperature is too high.	---

Signal Definitions (Continued)

2.4.16 GPIO Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
GPIO0	H1	D11	I/O	**GPIO Port 0.** Each signal is configured independently as an input or I/O, with or without static pull-up, and with either open-drain or totem-pole output type. A debouncer and an interrupt can be enabled or masked for each of signals GPIO[00:01] and [06:15] independently. **Note:** GPIO12, GPIO13, GPIO16 inputs: If GPIOx function is selected but not used, tie GPIOx low.	TRDE#
GPIO1	H2	D10			IOCS1#+TFTD12
	AL12	N30			AB1D+IOCS1#
GPIO6	AH3	D28			DTR2#/BOUT2+ IDE_IOR1#+ SDTEST5
GPIO7	AH4	C30			RTS2#+IDE_DACK1# +SDTEST0
GPIO8	AJ2	C31			CTS2#+IDE_DREQ1 +SDTEST4
GPIO9	AG4	C28			DCD2#+IDE_IOW1#+ SDTEST2
GPIO10	AJ1	B29			DSR2#+IDE_IORDY1 +SDTEST1
GPIO11	H30	AJ8			RI2#+IRQ15
GPIO12	AJ12	N29			AB2C
GPIO13	AL11	M29			AB2D
GPIO14	F1	D9			IOR#+DOCR#
GPIO15	G3	A8			IOW#+DOCW#
GPIO16	AL15	V31			PC_BEEP+ F_DEVSEL#
GPIO17	J4	A10			IOCS0#+TFTDCK
GPIO18	A28	AG1			DTR1#/BOUT1
GPIO19	H4	C9			INTC#+IOCHRDY
GPIO20	H3	A9			DOCCS#+TFTD0
	AJ13	N31			AB1C+DOCCS#
GPIO32	AJ11	M28	I/O	**GPIO Port 1.** Each signal is configured independently as an input or I/O, with or without static pull-up, and with either open-drain or totem-pole output type. A debouncer and an interrupt can be enabled or masked for each of signals GPIO[32:41] independently.	LAD0
GPIO33	AL10	L31			LAD1
GPIO34	AK10	L30			LAD2
GPIO35	AJ10	L29			LAD3
GPIO36	AL9	L28			LDRQ#
GPIO37	AK9	K31			LFRAME#
GPIO38/IRRX2	AJ9	K28			LPCPD#
GPIO39	AL8	J31			SERIRQ
GPIO40	A21	Y3			IDE_DATA8
GPIO41	C19	W4			IDE_DATA11

Signal Definitions (Continued)

2.4.17 Debug Monitoring Interface Signals

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
FPCICLK	U3	B18	O	**Fast-PCI Bus Monitoring Signals.** When enabled, this group of signals provides for monitoring of the internal Fast-PCI bus for debug purposes. To enable, pull up FPCI_MON (EBGA ball D3 / TEPBGA ball A4).	ACK#+TFTDE
F_AD7	U1	A18	O		PD7+TFTD13
F_AD6	V3	A20	O		PD6+TFTD1
F_AD5	V2	C19	O		PD5+TFTD11
F_AD4	V1	C18	O		PD4+TFTD10
F_AD3	W2	C20	O		PD3+TFTD9
F_AD2	W3	D20	O		PD2+TFTD8
F_AD1	Y1	A21	O		PD1+TFTD7
F_AD0	AA1	C21	O		PD0+TFTD6
F_C/BE3#	T4	C17	O		SLCT+TFTD15
F_C/BE2#	T3	D17	O		PE+TFTD14
F_C/BE1#	T1	B17	O		BUSY/WAIT#+ TFTD3
F_C/BE0#	AA3	D21	O		ERR#+TFTD4+
F_FRAME#	AB1	A22	O		STB#/WRITE#+ TFTD17
F_IRDY#	W1	B20	O		SLIN#/ASTRB#+ TFTD16
F_STOP#	AJ15	U29	O		AC97_RST#
F_DEVSEL#	AL15	V31	O		GPIO16+ PC_BEEP
F_GNT0#	AK14	U31	O		SDATA_IN
F_TRDY#	AL14	U30	O		BIT_CLK
INTR_O	AB2	D22	O	**CPU Core Interrupt.** When enabled, this signal provides for monitoring of the internal GX1 core INTR signal for debug purposes. To enable, pull up FPCI_MON (EBGA ball D3 / TEPBGA ball A4).	AFD#/DSTRB#+ TFTD2
SMI_O	Y3	B21	O	**System Management Interrupt.** This is the input to the GX1 core. When enabled, this signal provides for monitoring of the internal GX1 core SMI# signal for debug purposes. To enable, pull up FPCI_MON (EBGA ball D3 / TEPBGA ball A4).	INIT#+TFTD5+

Signal Definitions (Continued)

2.4.18 JTAG Interface Signals

Signal Name	Ball No.		Type	Description	Mux
	EBGA	TEPBGA			
TCK	AL4	E31	I	**JTAG Test Clock.** This signal has an internal weak pull-up resistor.	---
TDI	AK5	F29	I	**JTAG Test Data Input.** This signal has an internal weak pull-up resistor.	---
TDO	AH6	E30	O	**JTAG Test Data Output**	---
TMS	AJ5	F28	I	**JTAG Test Mode Select.** This signal has an internal weak pull-up resistor.	---
TRST#	AK4	E29	I	**JTAG Test Reset.** This signal has an internal weak pull-up resistor. For normal JTAG operation, this signal should be active at power-up. If the JTAG interface is not being used, this signal can be tied low.	---

Signal Definitions (Continued)

2.4.19 Test and Measurement Interface Signals

Signal Name	Ball No. EBGA	Ball No. TEPBGA	Type	Description	Mux
PLL6B	C28	AG4	I/O	**PLL6, PLL5 and PLL2 Bypass.** These signals are used for internal testing only. For normal operation leave unconnected.	TEST1
PLL5B	B29	AJ1	I/O		TEST2
PLL2B	D28	AH3	I/O		TEST0
GXCLK	AL16	V30	O	**GX Clock.** This signal is for internal testing only. For normal operation either program as FP_VDD_ON or leave unconnected.	FP_VDD_ON+ TEST3
TEST3	AL16	V30	O	**Internal Test Signals.** These signals are used for internal testing only. For normal operation, leave unconnected unless programmed as one of their muxed options.	FP_VDD_ON+ GXCLK
TEST2	B29	AJ1	O		PLL5B
TEST1	C28	AG4	O		PLL6B
TEST0	D28	AH3	O		PLL2B
GTEST	AL5	F30	I	**Global Test.** This signal is used for internal testing only. For normal operation this signal should be pulled down with 1.5 KΩ.	---
SDTEST5	AH3	D28	O	**Memory Internal Test Signals.** These signals are used for internal testing only. For normal operation, these signals should be programmed as one of their muxed options.	GPIO6+ DTR2#/BOUT2+ IDE_IOR1#
SDTEST4	AJ2	C31	O		GPIO8+CTS2#+ IDE_DREQ1
SDTEST3	AJ4	E28	O		SIN2
SDTEST2	AG4	C28	O		GPIO9+DCD2#+ IDE_IOW1#
SDTEST1	AJ1	B29	O		GPIO10+DSR2# +IDE_IORDY1
SDTEST0	AH4	C30	O		GPIO7+RTS2#+ IDE_DACK1#
TDP	AH5	D30	I/O	**Thermal Diode Positive / Negative**. These signals are for internal testing only. For normal operation leave unconnected.	---
TDN	AL3	D31	I/O		---

Signal Definitions (Continued)

2.4.20 Power, Ground and No Connections[1]

Signal Name	Ball No.		Type	Description
	EBGA	TEPBGA		
AV_{SSPLL2}	R3	C16	GND	**PLL2 Analog Ground Connection.**
AV_{SSPLL3}	E28	AK3	GND	**PLL3 Analog Ground Connection.**
V_{PLL2}	R1	A17	PWR	**3.3V PLL2 Analog Power Connection.** Low noise power for PLL2 and PLL5.
V_{PLL3}	C31	AJ4	PWR	**3.3V PLL3 Analog Power Connection.** Low noise power for PLL3, PLL4, and PLL6.
AV_{CCUSB}	AF4	D27	PWR	**3.3V Analog USB Power Connection.** Low noise power.
AV_{SSUSB}	AG1	C27	GND	**Analog USB Ground Connection.**
AV_{CCCRT}	L3, M1, N1	A12, C13, D15	PWR	**3.3V Analog CRT DAC Power Connections.** Low noise power.
AV_{SSCRT}	L1, N3, P3	B14, C14, C15	GND	**Analog CRT DAC Ground Connections**. Return current.
V_{CCCRT}	K3	D12	PWR	**1.8V / 2.1V CRT DAC Digital Power Connection.** Can be directly connected to V_{CORE} on PCB (printed circuit board).
V_{SSCRT}	K2	C12	GND	**CRT DAC Digital Ground Connection.** Can be directly connected to V_{SS} on PCB.
V_{BAT}	D30	AL3	PWR	**Battery.** Provides battery back-up to the RTC and ACPI registers, when V_{SB} is lower than the minimum value (see Table 8-2 on page 359). The ball is connected to the internal logic through a series resistor for UL protection.
V_{SB}	F29	AL5	PWR	**3.3V Standby Power Supply.** Provides power to the Real-Time Clock (RTC) and ACPI circuitry while the main power supply is turned off.
V_{SBL}	H28	AL6	PWR	**1.8V / 2.1V Standby Power Supply.** Provides power to the internal logic while the main power supply is turned off. This signal requires a 0.1 µF bypass capacitor to V_{SS}. This supply must be present when V_{SB} is present.
V_{CORE}	See Table 2-3 on page 32. (Total of 25)	See Table 2-5 on page 47. (Total of 28)	PWR	**1.8V / 2.1V Core Processor Power Connections.**
V_{IO}	See Table 2-3 on page 32. (Total of 32)	See Table 2-5 on page 47. (Total of 43)	PWR	**3.3V I/O Power Connections.**
V_{SS}	See Table 2-3 on page 32. (Total of 57)	See Table 2-5 on page 47. (Total of 92)	GND	**Ground Connections.**
NC	See Table 2-3 on page 32. (Total of 8)	See Table 2-5 on page 47. (Total of 8)	---	**No Connections.** These lines should be left disconnected. Connecting a pull-up/-down resistor or to an active signal could cause unexpected results and possible malfunctions.

1. All power sources must be connected, even if the function is not used

3.0 General Configuration Block

The General Configuration block includes registers for:

- Pin Multiplexing and Miscellaneous Configuration
- WATCHDOG Timer
- High-Resolution Timer
- Clock Generators

A selectable interrupt is shared by all these functions.

3.1 CONFIGURATION BLOCK ADDRESSES

Registers of the General Configuration block are I/O mapped in a 64-byte address range. These registers are physically connected to the internal Fast-PCI bus, but do not have a register block in PCI configuration space (i.e., they do not appear to software as PCI registers).

After system reset, the Base Address register is located at I/O address 02EAh. This address can be used only once. Before accessing any PCI registers, the BOOT code must program this 16-bit register to the I/O base address for the General Configuration block registers. All subsequent writes to this address, are ignored until system reset.

Note: Location of the General Configuration Block cannot be determined by software. See the *SC2200 Thin Client On a Chip device errata* document.

Reserved bits in the General Configuration block should read as written unless otherwise specified.

Table 3-1. General Configuration Block Register Summary

Offset	Width (Bits)	Type	Name	Reset Value	Reference
00h-01h	16	R/W	**WDTO.** WATCHDOG Timeout	0000h	Page 89
02h-03h	16	R/W	**WDCNFG.** WATCHDOG Configuration	0000h	Page 89
04h	8	R/WC	**WDSTS.** WATCHDOG Status	00h	Page 89
05h-07h	---	---	**RSVD.** Reserved	---	---
08h-0Bh	32	RO	**TMVALUE.** TIMER Value	xxxxxxxxh	Page 91
0Ch	8	R/W	**TMSTS.** TIMER Status	00h	Page 91
0Dh	8	R/W	**TMCNFG.** TIMER Configuration	00h	Page 91
0Eh-0Fh	---	---	**RSVD.** Reserved	---	---
10h	8	RO	**MCCM.** Maximum Core Clock Multiplier	Strapped Value	Page 95
11h	---	---	**RSVD.** Reserved	---	---
12h	8	R/W	**PPCR.** PLL Power Control	2Fh	Page 95
13h-17h	---	---	**RSVD.** Reserved	---	---
18h-1Bh	32	R/W	**PLL3C.** PLL3 Configuration	E1040005h	Page 96
1Ch-1Dh	---	---	**RSVD.** Reserved	---	---
1Eh-1Fh	16	R/W	**CCFC.** Core Clock Frequency Control	Strapped Value	Page 96
20h-2Fh	---	---	**RSVD.** Reserved	---	---
30h-33h	32	R/W	**PMR.** Pin Multiplexing Register	00000000h	Page 81
34h-37h	32	R/W	**MCR.** Miscellaneous Configuration Register	00000001h	Page 85
38h	8	R/W	**INTSEL.** Interrupt Selection	00h	Page 87
39h-3Bh	---	---	**RSVD.** Reserved	---	---
3Ch	8	RO	**IID.** IA On a Chip ID	xxh	Page 87
3Dh	8	RO	**REV.** Revision	xxh	Page 87
3Eh-3Fh	16	RO	**CBA.** Configuration Base Address	xxxxh	Page 87

3.2 MULTIPLEXING, INTERRUPT SELECTION, AND BASE ADDRESS REGISTERS

The registers described inTable 3-2 are used to determine general configuration for the SC2200. These registers also indicate which multiplexed signals are issued via balls from which more than one signal may be output. For more information about multiplexed signals and the appropriate configurations, see Section 2.1 "Ball Assignments" on page 20.

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers

Bit	Description

Offset 30h-33h **Pin Multiplexing Register - PMR (R/W)** **Reset Value: 00000000h**

This register configures pins with multiple functions. See Section 2.1 on page 20 for more information about multiplexing information.

31:30 — **Reserved:** Always write 0.

29 — **Test Signals.** Selects ball functions.

Ball # EBGA / TEPBGA	0: Internal Test Signals Name	0: Internal Test Signals Add'l Dependencies	1: Internal Test Signals Name	1: Internal Test Signals Add'l Dependencies
D28 / AH3	PLL2B	None	TEST0	None
C28 / AG4	PLL6B	None	TEST1	None
B29 / AJ1	PLL5B	None	TEST2	None
AL16 / V30	GXCLK	See PMR[23]	TEST3	PMR[23] = 0

28 — **Test Signals.** Selects ball function.

Ball # EBGA / TEPBGA	0: AC97 Signal Name	0: AC97 Signal Add'l Dependencies	1: Internal Test Signal Name	1: Internal Test Signal Add'l Dependencies
AJ4 / E28	SIN2	None	SDTEST3	See Note.

Note: If this bit is set, PMR[8] and PMR[18] must be set by software.

27 — **FPCI_MON (Fast-PCI Monitoring).** Selects Fast-PCI monitoring output signals instead of Parallel Port signals.

Fast-PCI monitoring output signals can be enabled in two ways: by setting this bit to 1 or by strapping FPCI_MON (EBGA ball D3 / TEPBGA ball A4) high. (The strapped value can be read back at MCR[30].) Listed below is how these two options work together and the signals that are enabled (enabling overrides add'l dependencies except FPCI_MON = 1). Note that the FPCI monitoring signals that are muxed with Audio signals are not enabled via this bit. They are only enabled using the strap option.

PMR[27]	FPCI_MON	
0	0	Disable all Fast-PCI monitoring signals
0	1	Enable all Fast-PCI monitoring signals
1	0	Enable Fast-PCI monitoring signals muxed with Parallel Port signals only
1	1	Enable all Fast-PCI monitoring signals

Ball # EBGA / TEPBGA	FPCI_MON Signal	Other Signal	Add'l Dependencies
U3 / B18	FPCICLK	ACK#+TFTDE	See PMR[23]
U1 / A18	F_AD7	PD7+TFTD13	See PMR[23]
V3 / A20	F_AD6	PD6+TFTD1	See PMR[23]
V2 / C19	F_AD5	PD5+TFT11	See PMR[23]
V1 / C18	F_AD4	PD4+TFTD10	See PMR[23]
W2 / C20	F_AD3	PD3+TFTD9	See PMR[23]
W3 / D20	F_AD2	PD2+TFTD8	See PMR[23]
Y1 / A21	F_AD1	PD1+TFTD7	See PMR[23]
AA1 / C21	F_AD0	PD0_TFTD5	See PMR[23]
T4 / C17	F_C/BE3#	SLCT+TFTD15	See PMR[23]
T3 / D17	F_C/BE2#	PE+TFTD14	See PMR[23]
T1 / B17	F_C/BE1#	BUSY/WAIT#+TFTD3	See PMR[23]
AA3 / D21	F_C/BE0#	ERR#+TFTD4	See PMR[23]
AB1 / A22	F_FRAME#	STB#/WRITE#+TFTD7	See PMR[23]
W1 / B20	F_IRDY#	SLIN#/ASTRB#+TFTD16	See PMR[23]
AB2 / D22	INTR_O	AFD#/DSTRB#+TFTD2	See PMR[23]
Y3 / B21	SMI_O	INIT#+TFTD5	See PMR[23]
AL15 / V31	F_DEVSEL#	GPIO16+PC_BEEP	FPCI_MON = 1 and see PMR[0]
AJ15 / U29	F_STOP#	AC97_RST#	FPCI_MON = 1
AK14 / U31	F_GNT0#	SDATA_IN	FPCI_MON = 1
AL14 / U30	F_TRDY#	BIT_CLK	FPCI_MON = 1

26 — **Note:** **Reserved:** Always write 0.

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
25	**AC97CKEN (Enable AC97_CLK Output).** This bit enables the output drive of AC97_CLK (EBGA ball AJ14 / TEPBGA ball P31). 0: AC97_CLK output is HIZ. 1: AC97_CLK output is enabled.
24	**TFTIDE (TFT/IDE).** Determines whether certain balls are used for TFT signals or for IDE signals. Note that there are no additional dependencies.

Ball # EBGA / TEPBGA	**0: IDE Signals Name**	**1: CRT, GPIO and TFT Signals Name**
A26 / AD3	IDE_ADDR0	TFTD3
C26 / AE1	IDE_ADDR1	TFTD2
C17 / U2	IDE_ADDR2	TFTD4
B24 / AC3	IDE_DATA0	TFTD6
A24 / AC1	IDE_DATA1	TFTD16
D23 / AC2	IDE_DATA2	TFTD14
C23 / AB4	IDE_DATA3	TFTD12
B23 / AB1	IDE_DATA4	FP_VDD_ON
A23 / AA4	IDE_DATA5	CLK27M
C22 / AA3	IDE_DATA6	IRQ9
B22 / AA2	IDE_DATA7	INTD#
A21 / Y3	IDE_DATA8	GPIO40
C20 / Y2	IDE_DATA9	DDC_SDA
A20 / Y1	IDE_DATA10	DDC_SCL
C19 / W4	IDE_DATA11	GPIO41
B19 / W3	IDE_DATA12	TFTD13
A19 / V3	IDE_DATA13	TFTD15
C18 / V2	IDE_DATA14	TFTD17
B18 / V1	IDE_DATA15	TFTD7
A27 / AF2	IDE_CS0#	TFTD5
C16 / P2	IDE_CS1#	TFTDE
C21 / Y4	IDE_IOR0#	TFTD10
D24 / AD2	IDE_IOW0#	TFTD9
C24 / AC4	IDE_DREQ0	TFTD8
C25 / AD4	IDE_DACK0#	TFTD0
A22 / AA1	IDE_RST#	TFTDCK
A25 / AD1	IDE_IORDY0	TFTD11
D25 / AF1	IRQ14	TFTD1

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>23</td><td>
TFTPP (TFT/Parallel Port). Determines whether certain balls are used for TFT or PP/ACB1/FPCI. This bit is set to 1 at power-on if the TFT_PRSNT strap (EBGA ball AK13 / TEPBGA ball P29) is pulled high.
<table>
<tr><th>Ball #
EBGA / TEPBGA</th><th>0: PP/ACB1/FPCI
Name</th><th>Add'l Dependencies</th><th>1: TFT
Name</th><th>Add'l Dependencies</th></tr>
<tr><td>H2 / D10</td><td>GPIO1
IOCS1#</td><td>PMR[13] = 0
PMR[13] = 1</td><td>TFTD12</td><td>None</td></tr>
<tr><td>H3 / A9</td><td>GPIO20
DOCCS#</td><td>PMR[7] = 0
PMR[7] = 1</td><td>TFTD0</td><td>None</td></tr>
<tr><td>J4 / A10</td><td>GPIO17
IOCS0#</td><td>PMR[5] = 0
PMR[5] = 1</td><td>TFTDCK</td><td>None</td></tr>
<tr><td>T1 / B17</td><td>BUSY/WAIT#
F_C/BE1#</td><td>Note 1
Note 2</td><td>TFTD3</td><td>None</td></tr>
<tr><td>T3 / D17</td><td>PE
F_C/BE2#</td><td>Note 1
Note 2</td><td>TFTD14</td><td>Note 1</td></tr>
<tr><td>T4 / C17</td><td>SLCT
F_C/BE3#</td><td>Note 1
Note 2</td><td>TFTD15</td><td>Note 1</td></tr>
<tr><td>U1 / A18</td><td>PD7
F_AD7</td><td>Note 1
Note 2</td><td>TFTD13</td><td>Note 1</td></tr>
<tr><td>U3 / B18</td><td>ACK#
FPCICLK</td><td>Note 1
Note 2</td><td>TFTDE</td><td>Note 1</td></tr>
<tr><td>V1 / C18</td><td>PD4
F_AD4</td><td>Note 1
Note 2</td><td>TFTD10</td><td>Note 1</td></tr>
<tr><td>V2 / C19</td><td>PD5
F_AD5</td><td>Note 1
Note 2</td><td>TFTD11</td><td>Note 1</td></tr>
<tr><td>V3 / A20</td><td>PD6
F_AD6</td><td>Note 1
Note 2</td><td>TFTD1</td><td>Note 1</td></tr>
<tr><td>W1 / B20</td><td>SLIN#/ASTRB#
F_IRDY#</td><td>Note 1
Note 2</td><td>TFTD16</td><td>Note 1</td></tr>
<tr><td>W2 / C20</td><td>PD3
F_AD3</td><td>Note 1
Note 2</td><td>TFTD9</td><td>Note 1</td></tr>
<tr><td>W3 / D20</td><td>PD2
F_AD2</td><td>Note 1
Note 2</td><td>TFTD8</td><td>Note 1</td></tr>
<tr><td>Y1 / A21</td><td>PD1
F_AD1</td><td>Note 1
Note 2</td><td>TFTD7</td><td>Note 1</td></tr>
<tr><td>Y3 / B21</td><td>INIT#
SMI_O</td><td>Note 1
Note 2</td><td>TFTD5</td><td>Note 1</td></tr>
<tr><td>AA1 / C21</td><td>PD0
F_AD0</td><td>Note 1
Note 2</td><td>TFTD6</td><td>Note 1</td></tr>
<tr><td>AA3 / D21</td><td>ERR#
F_C/BE0#</td><td>Note 1
Note 2</td><td>TFTD4</td><td>Note 1</td></tr>
<tr><td>AB1 / A22</td><td>STB#/WRITE#
F_FRAME#</td><td>Note 1
Note 2</td><td>TFTD17</td><td>None</td></tr>
<tr><td>AB2 / D22</td><td>AFD#/DSTRB#
INTR_O</td><td>Note 1
Note 2</td><td>TFTD2</td><td>Note 1</td></tr>
<tr><td>AJ13 / N31</td><td>AB1C</td><td>None</td><td>GPIO20
DOCCS#</td><td>PMR[7] = 0
PMR[7] = 1</td></tr>
<tr><td>AL12 / N30</td><td>AB1D</td><td>None</td><td>GPIO1
IOCS1#</td><td>PMR[13] = 0
PMR[13] = 1</td></tr>
<tr><td>AL16 / V30</td><td>GXCLK
TEST3</td><td>PMR[29] = 0
PMR[29] = 1</td><td>FP_VDD_ON</td><td>None</td></tr>
</table>
Note: 1. PMR[27] = 0 and FPCI_MON = 0

2. PMR[27] = 1 or FPCI_MON = 1

3. ACCESS.bus interface 1 is not available if PMR[23] = 1.

4. If FPCI_MON strap is enabled, the TFT_PRSNT strap should pulled low.
</td></tr>
<tr><td>22</td><td>RSVD (Reserved). Must be set equal to PMR[14] (LPCSEL). The LPC_ROM strap (EBGA ball E4 / TEPBGA ball D6) determines the power-on reset (POR) state of PMR[14] and PMR[22].</td></tr>
</table>

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description				
21	**IOCSEL (Select I/O Commands).** Selects ball functions.				
	Ball #	**0: I/O Command Signals**		**1: GPIO Signals**	
	EBGA / TEPBGA	**Name**	**Add'l Dependencies**	**Name**	**Add'l Dependencies**
	F1 / D9	IOR#	PMR[2] = 0	GPIO14	PMR[2] = 1
		DOCR#	PMR[2] = 1	Undefined	PMR[2] = 0
	G3 / A8	IOW#	PMR[2] = 0	GPIO15	PMR[2] = 1
		DOCW#	PMR[2] = 1	Undefined	PMR[2] = 0
20	**Reserved.** Must be set to 0.				
19	**AB2SEL (Select ACCESS.bus 2).** Selects ball functions.				
	Ball #	**0: GPIO Signals**		**1: ACCESS.bus 2 Signals**	
	EBGA / TEPBGA	**Name**	**Add'l Dependencies**	**Name**	**Add'l Dependencies**
	AJ12 / N29	GPIO12	None	AB2C	None
	AL11 / M29	GPIO13	None	AB2D	None
18	**SP2SEL (Select SP2 Additional Pins).** Selects ball functions.				
	Ball #	**0: GPIO, IDE Signals**		**1: Serial Port Signals**	
	EBGA / TEPBGA	**Name**	**Add'l Dependencies**	**Name**	**Add'l Dependencies**
	AH3 / D28	GPIO6	PMR[8] = 0	DTR2#/BOUT2	PMR[8] = 0
		IDE_IOR1#	PMR[8] = 1	SDTEST5	PMR[8] = 1
	AG4 / C28	GPIO9	PMR[8] = 0	DCD2#	PMR[8] = 0
		IDE_IOW1#	PMR[8] = 1	SDTEST2	PMR[8] = 1
	AJ1 / B29	GPIO10	PMR[8] = 0	DSR2#	PMR[8] = 0
		IDE_IORDY1	PMR[8] = 1	SDTEST1	PMR[8] = 1
	H30 / AJ8	GPIO11	PMR[8] = 0	RI2#	PMR[8] = 0
		IRQ15	PMR[8] = 1	Undefined	PMR[8] = 1
17	**SP2CRSEL (Select SP2 Flow Control).** Selects ball functions.				
	Ball #	**0: GPIO, IDE Signals**		**1: Serial Port Signals**	
	EBGA / TEPBGA	**Name**	**Add'l Dependencies**	**Name**	**Add'l Dependencies**
	AH4 / C30	GPIO7	PMR[8] = 0	RTS2#	PMR[8] = 0
		IDE_DACK1#	PMR[8] = 1	SDTEST0	PMR[8] = 1
	AJ2 / C31	GPIO8	PMR[8] = 0	CTS2#	PMR[8] = 0
		IDE_DREQ1	PMR[8] = 1	SDTEST4	PMR[8] = 1
16	**SP1SEL (Select SP1 Additional Pin)**. Selects ball function.				
	Ball #	**0: GPIO Signal**		**1: Serial Port Signal**	
	EBGA / TEPBGA	**Name**	**Add'l Dependencies**	**Name**	**Add'l Dependencies**
	A28 / AG1	GPIO18	None	DTR1#/BOUT1	None
15	**RSVD (Reserved).** Write to 0.				
14	**LPCSEL (Select LPC Bus).** Selects ball functions. The LPC_ROM strap (EBGA ball E4 / TEPBGA ball D6) determines the power-on reset (POR) state of PMR[14] and PMR[22].				
	Ball #	**0: GPIO Signals**		**1: LPC Signals**	
	EBGA / TEPBGA	**Name**	**Add'l Dependencies**	**Name**	**Add'l Dependencies**
	AJ11 / M28	GPIO32	PMR[22] = 0	LAD0	PMR[22] = 1
	AL10 / L31	GPIO33	PMR[22] = 0	LAD1	PMR[22] = 1
	AK10 / L30	GPIO34	PMR[22] = 0	LAD2	PMR[22] = 1
	AJ10 / L29	GPIO35	PMR[22] = 0	LAD3	PMR[22] = 1
	AL9 / L28	GPIO36	PMR[22] = 0	LDRQ#	PMR[22] = 1
	AK9 / K31	GPIO37	PMR[22] = 0	LFRAME#	PMR[22] = 1
	AJ9 / K28	GPIO38/IRRX2	PMR[22] = 0	LPCPD#	PMR[22] = 1
	AL8 / J31	GPIO39	PMR[22] = 0	SERIRQ	PMR[22] = 1
13	**IOCS1SEL (Select IOCS1).** Selects ball functions for IOCS1# or GPIO1. Works in conjunction with PMR[23], see PMR[23] for definition.				

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
12	**TRDESEL (Select TRDE#).** Selects ball function. **Ball # EBGA / TEPBGA** — **0: Sub-ISA Signal Name** — **Add'l Dependencies** — **1: GPIO Signal Name** — **Add'l Dependencies** H1 / D11 — TRDE# — None — GPIO0 — None
11	**EIDE (Enable IDE Outputs).** This bit enables IDE output signals. 0: IDE signals are HIZ. Other signals multiplexed on the same balls are HIZ until this bit is set. (without regard to bit 24 of this register). This bit does not control IDE channel 1 control signals selected by bit 8 of this register. 1: Signals are enabled.
10	**ETFT (Enable TFT Outputs).** This bit enables TFT output signals, that are multiplexed with the Parallel Port and controlled by PMR[23]. 0: Signals TFTD[17:0], TFTDE and TFTDCK are set to 0. 1: Signals TFTD[17:0], TFTDE and TFTDCK are enabled. **Note:** TFTDCK that is multiplexed on IDE_RST# (EBGA ball A22 / TEPBGA ball AA1) is also enabled by this bit.
9	**IOCHRDY (Select IOCHRDY).** Selects ball function. **Ball # EBGA / TEPBGA** — **0: PCI, GPIO Signal Name** — **Add'l Dependencies** — **1: Sub-ISA Signal Name** — **Add'l Dependencies** H4 / C9 — GPIO19 — PMR[4] = 0 — IOCHRDY — PMR[4] = 1 INTC# — PMR[4] = 1 — Undefined — PMR[4] = 0
8	**IDE1SEL (Select IDE Channel 1).** Selects IDE Channel 1 or GPIO ball functions. Works in conjunction with PMR[18] and PMR[17], see PMR[18] and PMR[17] for definitions.
7	**DOCCSSEL (Select DOCCS#).** Selects DOCCS# or GPIO20 ball functions. Works in conjunction with PMR[23], see PMR[23] for definition.
6	**SP3SEL (Select UART3).** Selects ball functions. **Ball # EBGA / TEPBGA** — **0: IR Signals Name** — **Add'l Dependencies** — **1: Serial Port Signals Name** — **Add'l Dependencies** J28 / AK8 — IRRX1 — None — SIN3 — None J3 / C11 — IRTX — None — SOUT3 — None
5	**IOCS0SEL (Select IOCS0#).** Selects ball function. Works in conjunction with PMR[23], see PMR[23] for definition.
4	**INTCSEL (Select INTC#).** Selects ball function. Works in conjunction with PMR[9], see PMR[9] for definition.
3	**Reserved.** Write as read.
2	**DOCWRSEL (Select DiskOnChip and NAND Flash Command Lines).** Selects ball functions. Works in conjunction with PMR[21], see PMR[21] for definition.
1	**Reserved.** Write as read.
0	**PCBEEPSEL (Select PC_BEEP)**. Selects ball function. **Ball # EBGA / TEPBGA** — **0: GPIO Signal Name** — **Add'l Dependencies** — **1: Audio Signal Name** — **Add'l Dependencies** AL15 / V31 — GPIO16 — FPCI_MON = 0 — PC_BEEP — FPCI_MON] = 0 F_DEVSEL# — FPCI_MON = 1 — F_DEVSEL# — FPCI_MON = 1

Offset 34h-37h **Miscellaneous Configuration Register - MCR (R/W)** **Reset Value: 0000001h**

Power-on reset value: The BOOT16 strap pin selects "Enable 16-Bit Wide Boot Memory".

31	**DID0 (EBGA Ball D4 / TEPBGA Ball C5) Strap Status. (Read Only)** Represents the value of the strap that is latched after power-on reset. Read in conjunction with bit 29.
30	**FPCI_MON (EBGA Ball D3 / TEPBGA Ball A4) Strap Status. (Read Only)** Represents the value of the strap that is latched after power-on reset. Indicates if Fast-PCI monitoring output signals (instead of Parallel Port and some audio signals) are enabled. The state of this bit along with PMR[27] control the Fast-PCI monitoring function. See PMR[27] definition.
29	**DID1 (EBGA Ball D2 / TEPBGA Ball C6) Strap Status. (Read Only)** Represents the value of the strap that is latched after power-on reset. Read in conjunction with bit 31.
28:20	**Reserved**
19:18	**Reserved.** Write as 0.
17	**HSYNC Timing.** HSYNC timing control for TFT. 0: HSYNC timing suited for CRT. 1: HSYNC timing suited for TFT.

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
16	**Delay HSYNC.** HSYNC delay by two TFT clock cycles. 0: There is no delay on HSYNC. 1: HYSNC is delayed twice by rising edge of TFT clock. Enables delay between VSYNC and HSYNC suited for TFT display.
15	**Reserved.** Write as read.
14	**IBUS16 (Invert BUS16).** This bit inverts the meaning of MCR[3] (bit 3 of this register). 0: BUS16 is as described for MCR[3]. 1: BUS16 meaning is inverted: if MCR[3] = 0, ROMCS# access is 16 bits wide; if MCR[3] = 1, ROMCS# access is 8 bits wide.
13	**Reserved.** Must be set to 0.
12	**IO1ZWS (Enable ZWS# for IOCS1# Access).** This bit enables internal activation of ZWS# (Zero Wait States) control for IOCS1# access. 0: ZWS# is not active for IOCS1# access. 1: ZWS# is active for IOCS1# access.
11	**IO0ZWS (Enable ZWS# for IOCS0# Access).** This bit enables internal activation of ZWS# (Zero Wait States) control for IOCS0# access. 0: ZWS# is not active for IOCS0# access. 1: ZWS# is active for IOCS0# access.
10	**DOCZWS (Enable ZWS# for DOCCS# Access).** This bit enables internal activation of ZWS# (Zero Wait States) control for DOCCS# access. 0: ZWS# is not active for DOCCS# access. 1: ZWS# is active for DOCCS# access.
9	**ROMZWS (Enable ZWS# for ROMCS# Access).** This bit enables internal activation of ZWS# (Zero Wait States) control for ROMCS# access. 0: ZWS# is not active for ROMCS# access. 1: ZWS# is active for ROMCS# access.
8	**IO1_16 (Enable 16-Bit Wide IOCS1# Access).** This bit enables the16-line access to IOCS1# in the Sub-ISA interface. 0: 8-bit wide IOCS1# access is used. 1: 16-bit wide IOCS1# access is used.
7	**IO0_16 (Enable 16-Bit Wide IOCS0# Access).** This bit enables the 16-line access to IOCS0# in the Sub-ISA interface. 0: 8-bit wide IOCS0# access is used. 1: 16-bit wide IOCS0# access is used.
6	**DOC16 (Enable 16-Bit Wide DOCCS# Access).** This bit enables the 16-line access to DOCCS# in the Sub-ISA interface. 0: 8-bit wide DOCCS# access is used. 1: 16-bit wide DOCCS# access is used.
5	**Reserved.** Write as read.
4	**IRTXEN (Infrared Transmitter Enable).** This bit enables drive of Infrared transmitter output. 0: IRTX+SOUT3 line (EBGA ball J3 / TEPBGA ball C11) is HIZ. 1: IRTX+SOUT3 line (EBGA ball J3 / TEPBGA ball C11) is enabled.
3	**BUS16 (16-Bit Wide Boot Memory). (Read Only)** This bit reports the status of the BOOT16 strap (EBGA ball G4 / TEPBGA ball C8). If the BOOT16 strap is pulled high, at reset 16-bit access to ROM in the Sub-ISA interface is enabled. MCR[14] = 1 inverts the meaning of this register. 0: 8-bit wide ROM. 1: 16-bit wide ROM.
2:1	**Reserved.** Write as read.

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
0	**SDBE0 (Slave Disconnect Boundary Enable).** Works in conjunction with the GX1 module's PCI Control Function 2 Register (Index 41h), bit 1 (SDBE1). Sets boundaries for when the GX1 module is a PCI slave. SDBE[1:0] 00: Read and Write disconnect on boundaries set by bits [3:2] of the GX1 module's PCI Control Function 2 register (Index 41h). 01: Write disconnects on boundaries set by bits [3:2] of the GX1 module's PCI Control Function 2 register. Read disconnects on cache line boundary of 16 bytes. 1x: Read and Write disconnect on cache line boundary of 16 bytes. This bit is reset to 1. All PCI bus masters (including SC2200's on-chip PCI bus masters, e.g., the USB Controller) must be disabled while modifying this bit. When accessing this register while any PCI bus master is enabled, use read-modify-write to ensure these bit contents are unchanged.
Offset 38h	**Interrupt Selection Register - INTSEL (R/W)** — **Reset Value: 00h** This register selects the IRQ signal of the combined WATCHDOG and High-Resolution timer interrupt. This interrupt is shareable with other interrupt sources.
7:4	**Reserved.** Write as read.
3:0	**CBIRQ.** Configuration Block Interrupt. 0000: Disable — 0100: IRQ4 — 1000: IRQ8# — 1100: IRQ12 0001: IRQ1 — 0101: IRQ5 — 1001: IRQ9 — 1101: Reserved 0010: Reserved — 0110: IRQ6 — 1010: IRQ10 — 1110: IRQ14 0011: IRQ3 — 0111: IRQ7 — 1011: IRQ11 — 1111: IRQ15
Offset 39h-3Bh	**Reserved - RSVD**
Offset 3Ch	**IA On a Chip Identification Number Register - IID (RO)** — **Reset Value: xxh** This register identifies the IA On a Chip device. SC2200 = 04h.
Offset 3Dh	**Revision Register - REV (RO)** — **Reset Value: xxh** This register identifies the device revision. See device errata for value.
Offset 3Eh-3Fh	**Configuration Base Address Register - CBA (RO)** — **Reset Value: xxh** This register sets the base address of the Configuration block.
15:6	**Configuration Base Address.** These bits are the high bits of the Configuration Base Address.
5:0	**Configuration Base Address.** These bits are the low bits of the Configuration Base Address. These bits are set to 0.

3.3 WATCHDOG

The SC2200 includes a WATCHDOG function to serve as a fail-safe mechanism in case the system becomes hung. When triggered, the WATCHDOG mechanism returns the system to a known state by generating an interrupt, an SMI, or a system reset (depending on configuration).

3.3.1 Functional Description

WATCHDOG is enabled when the WATCHDOG Timeout (WDTO) register (Offset 00h) is set to a non-zero value. The WATCHDOG timer starts with this value and counts down until either the count reaches 0, or a trigger event restarts the count (with the WDTO register value).

The WATCHDOG timer is restarted in any of the following cases:

- The WDTO register is set with a non-zero value.
- The WATCHDOG timer reaches 0 and the WATCHDOG Overflow bit, WDOVF (Offset 04h[0]), is 0.

The WATCHDOG function is disabled in any of the following cases:

- System reset occurs.
- The WDTO register is set to 0.
- The WDOVF bit is already 1 when the timer reaches 0.

3.3.1.1 WATCHDOG Timer

The WATCHDOG timer is a 16-bit down counter. Its input clock is a 32 KHz clock divided by a predefined value (see WDPRES field, Offset 02h[3:0]). The 32 KHz input clock is enabled when either:

- The GX1 module's internal SUSPA# signal is 1.

or

- The GX1 module's internal SUSPA# signal is 0 and the WD32KPD bit (Offset 02h[8]) is 0.

The 32 KHz input clock is disabled, when:

- The GX1 module's internal SUSPA# signal is 0 and the WD32KPD bit is 1.

For more information about signal SUSPA#, refer to the *GX1 Processor Series Datasheet*.

When the WATCHDOG timer reaches 0:

- If the WDOVF bit in the WDSTS register (Offset 04h[0]) is 0, an interrupt, an SMI or a system reset is generated, depending on the value of the WDTYPE1 field in the WDCNFG register (Offset 02h[5:4]).
- If the WDOVF bit in the WDSTS register is already 1 when the WATCHDOG timer reaches 0, an interrupt, an SMI or a system reset is generated according to the WDTYPE2 field (Offset 02h[7:6]), and the timer is disabled. The WATCHDOG timer is re-enabled when a non-zero value is written to the WDTO register (Offset 00h).

The interrupt or SMI is de-asserted when the WDOVF bit is set to 0. The reset generated by the WATCHDOG functionis used to trigger a system reset via the Core Logic module. The value of the WDOVF bit, the WDTYPE1 field, and the WDTYPE2 field are not affected by a system reset (except when generated by power-on reset).

The SC2200 also allows no action to be taken when the timer reaches 0 (according to WDTYPE1 field and WDTYPE2 field). In this case only the WDOVF bit is set to 1.

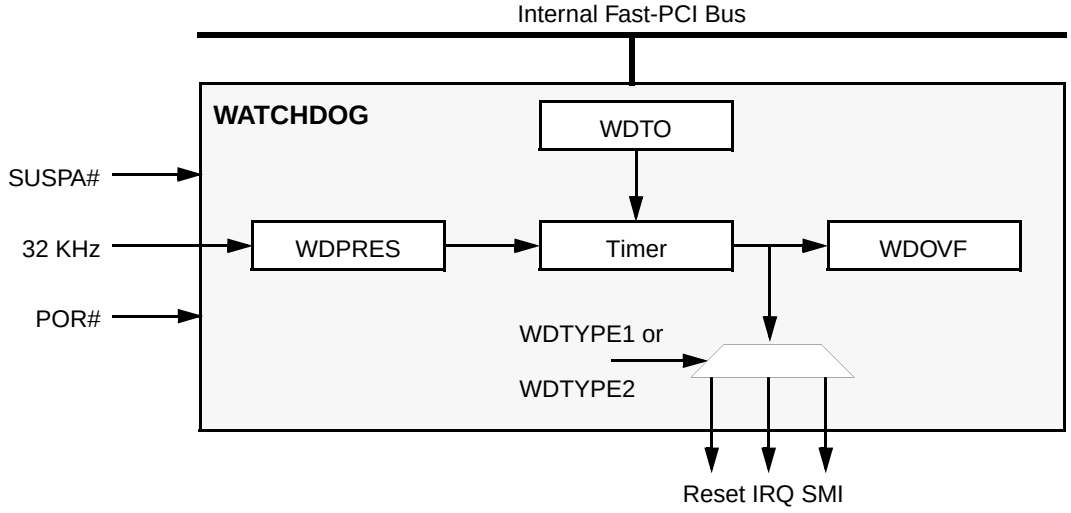


Figure 3-1. WATCHDOG Block Diagram

General Configuration Block (Continued)

WATCHDOG Interrupt

The WATCHDOG interrupt (if configured and enabled) is routed to an IRQ signal. The IRQ signal is programmable via the INTSEL register (Offset 38h, described in Table 3-2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81). The WATCHDOG interrupt is a shareable, active low, level interrupt.

WATCHDOG SMI

The WATCHDOG SMI is recognized by the Core Logic module as internal input signal EXT_SMI0#. To use the WATCHDOG SMI, Core Logic registers must be configured appropriately.

3.3.2 WATCHDOG Registers

Table 3-3 describes the WATCHDOG registers.

3.3.2.1 Usage Hints

- SMM code should set bit 8 of the WDCNFG register to 1 when entering ACPI C3 state, if the WATCHDOG timer is to be suspended. If this is not done, the WATCHDOG timer is functional during C3 state.
- SMM code should set bit 8 of the WDCNFG register to 1, when entering ACPI S1 and S2 states if the WATCHDOG timer is to be suspended. If this is not done, the WATCHDOG timer is functional during S1 and S2 states.

Table 3-3. WATCHDOG Registers

Bit	Description
Offset 00h-01h **WATCHDOG Timeout Register - WDTO (R/W)** **Reset Value: 0000h** This register specifies the programmed WATCHDOG timeout period.	
15:0	Programmed timeout period.
Offset 02h-03h **WATCHDOG Configuration Register - WDCNFG (R/W)** **Reset Value: 0000h** This register selects the signal to be generated when the timer reaches 0, whether or not to disable the 32 KHz input clock during low power states, and the prescaler value of the clock input.	
15:9	**Reserved.** Write as read.
8	**WD32KPD (WATCHDOG 32 KHz Power Down).** 0: 32 KHz clock is enabled. 1: 32 KHz clock is disabled, when the GX1 module asserts its internal SUSPA# signal. This bit is cleared to 0, when POR# is asserted or when the GX1 module de-asserts its internal SUSPA# signal (i.e., on SUSPA# rising edge). See Section 3.3.2.1 "Usage Hints" on page 89.
7:6	**WDTYPE2 (WATCHDOG Event Type 2)**. 00: No action 01: Interrupt 10: SMI 11: System reset This field is reset to 0 when POR# is asserted. Other system resets do not affect this field.
5:4	**WDTYPE1 (WATCHDOG Event Type 1).** 00: No action 01: Interrupt 10: SMI 11: System reset This field is reset to 0 when POR# is asserted. Other system resets do not affect this field.
3:0	**WDPRES (WATCHDOG Timer Prescaler).** Divide 32 KHz by: 0000: 1 0100: 16 1000: 256 1100: 4096 0001: 2 0101: 32 1001: 512 1101: 8192 0010: 4 0110: 64 1010: 1024 1110: Reserved 0011: 8 0111: 128 1011: 2048 1111: Reserved
Offset 04h **WATCHDOG Status Register - WDSTS (R/WC)** **Reset Value: 00h** This register contains WATCHDOG status information.	
7:4	**Reserved.** Write as read.
3	**WDRST (WATCHDOG Reset Asserted) (Read Only)** This bit is set to 1 when WATCHDOG Reset is asserted. It is set to 0 when POR# is asserted, or when the WDOVF bit is set to 0.
2	**WDSMI (WATCHDOG SMI Asserted.) (Read Only)** This bit is set to 1 when WATCHDOG SMI is asserted. It is set to 0 when POR# is asserted, or when the WDOVF bit is set to 0.

Table 3-3. WATCHDOG Registers (Continued)

Bit	Description
1	**WDINT (WATCHDOG Interrupt Asserted, Read Only).** This bit is set to 1 when the WATCHDOG Interrupt is asserted. It is set to 0 when POR# is asserted, or when the WDOVF bit is set to 0.
0	**WDOVF (WATCHDOG Overflow).** This bit is set to 1 when the WATCHDOG Timer reaches 0. It is set to 0 when POR# is asserted, or when a 1 is written to this bit by software. Other system reset sources do not affect this bit.
Offset 05h-07h	**Reserved - RSVD**

3.4 HIGH-RESOLUTION TIMER

The SC2200 provides an accurate time value that can be used as a time stamp by system software. This time is called the High-Resolution Timer. The length of the timer value can be extended via software. It is normally enabled while the system is in the C0 and C1 states. Optionally, software can be programmed to enable use of the High-Resolution Timer during C3 state and/or S1 state as well. In all other power states the High-Resolution Timer is disabled.

3.4.1 Functional Description

The High-Resolution Timer is a 32-bit free-running count-up timer that uses the oscillator clock or the oscillator clock divided by 27. Bit TMCLKSEL of the TMCNFG register (Offset 0Dh[1]) can be set via software to determine which clock should be used for the High-Resolution Timer.

When the most significant bit (bit 31) of the timer changes from 1 to 0, bit TMSTS of the TMSTS register (Offset 0Ch[0]) is set to 1. When both bit TMSTS and bit TMEN (Offset 0Dh[0]) are 1, an interrupt is asserted. Otherwise, the interrupt is de-asserted. This interrupt enables software emulation of a larger timer.

The High-Resolution Timer interrupt is routed to an IRQ signal. The IRQ signal is programmable via the INTSEL register (Offset 38h). For more information about this register, see section Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81.

System software uses the read-only TMVALUE register (Offset 08h[31:0]) to read the current value of the timer. The TMVALUE register has no default value.

The input clock (derived from the 27 MHz crystal oscillator) is enabled when:

- The GX1 module's internal SUSPA# signal is 1.

or

- The GX1 module's internal SUSPA# signal is 0 and bit TM27MPD (Offset 0Dh[2]) is 0.

The input clock is disabled, when the GX1 module's internal SUSPA# signal is 0 and the TM27MPD bit is 1.

For more information about signal SUSPA# see Section 3.4.2.1 "Usage Hints" on page 90 and the *GX1 Processor Series Datasheet*.

The High-Resolution Timer function resides on the internal Fast-PCI bus and its registers are in General Configuration Block address space. Only one complete register should be accessed at-a-time (e.g., DWORD access should be used for DWORD wide registers and byte access should be used for byte-wide registers).

3.4.2 High-Resolution Timer Registers

Table 3-4 on page 91 describes the registers for the High-Resolution Timer (TIMER).

3.4.2.1 Usage Hints

- SMM code should set bit 2 of the TMCNFG register to 1 when entering ACPI C3 state if the High-Resolution Timer should be disabled. If this is not done, the High-Resolution Timer is functional during C3 state.
- SMM code should set bit 2 of the TMCNFG register to 1 when entering ACPI S1 state if the High-Resolution Timer should be disabled. If this is not done, the High-Resolution Timer is functional during S1 state.

Table 3-4. High-Resolution Timer Registers

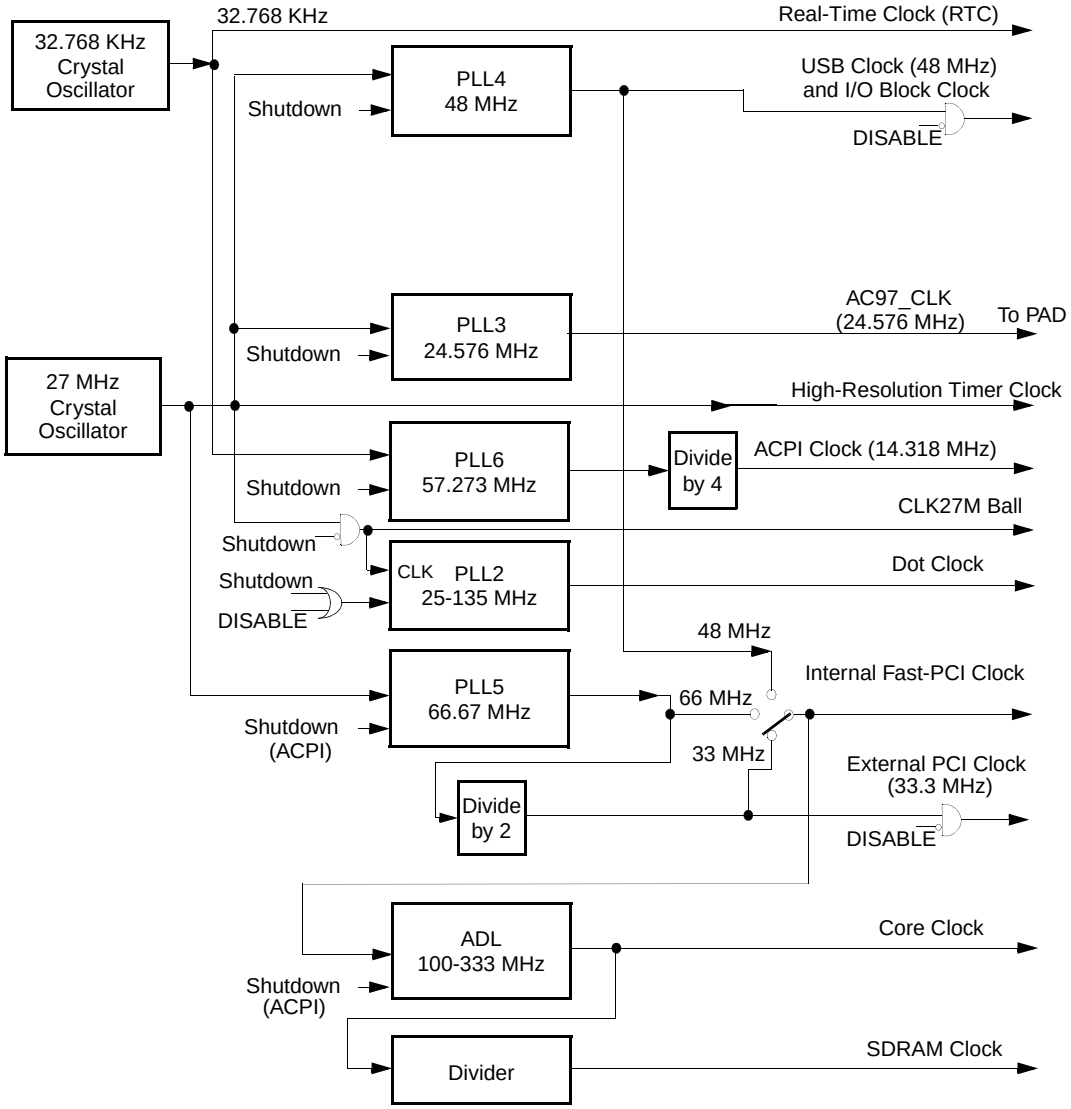
Bit	Description
Offset 08h-0Bh	**TIMER Value Register - TMVALUE (RO)** **Reset Value: xxxxxxxxh**
This register contains the current value of the High-Resolution Timer.	
31:0	**Current Timer Value.**
Offset 0Ch	**TIMER Status Register - TMSTS (R/W)** **Reset Value: 00h**
This register supplies the High-Resolution Timer status information.	
7:1	**Reserved.**
0	**TMSTS (TIMER Status).** This bit is set to 1 when the most significant bit (bit 31) of the timer changes from 1 to 0. It is cleared to 0 upon system reset or when 1 is written by software to this bit.
Offset 0Dh	**TIMER Configuration Register - TMCNFG (R/W)** **Reset Value: 00h**
This register enables the High-Resolution Timer interrupt; selects the Timer clock; and disables the 27 MHz internal clock during low power states.	
7:3	**Reserved.**
2	**TM27MPD (TIMER 27 MHz Power Down).** This bit is cleared to 0 when POR# is asserted or when the GX1 module de-asserts its internal SUSPA# signal (i.e., on SUSPA# rising edge). See Section 3.4.2.1 "Usage Hints" on page 90. 0: 27 MHz input clock is enabled. 1: 27 MHz input clock is disabled when the GX1 module asserts its internal SUSPA# signal.
1	**TMCLKSEL (TIMER Clock Select).** 0: Count-up timer uses the oscillator clock divided by 27. 1: Count-up timer uses the oscillator clock, 27 MHz clock.
0	**TMEN (TIMER Interrupt Enable).** 0: High-Resolution Timer interrupt is disabled. 1: High-Resolution Timer interrupt is enabled.
Offset 0Eh-0Fh	**Reserved - RSVD**

General Configuration Block (Continued)

3.5 CLOCK GENERATORS AND PLLS

This section describes the registers for the clocks required by the GX1 module, Core Logic module, and the Video Processor, and how these clocks are generated. See Figure 3-2 for a clock generation diagram.

The clock generators are based on 32.768 KHz and 27.000 MHz crystal oscillators. The 32.768 KHz crystal oscillator is described in Section 4.5.2 "RTC Clock Generation" on page 114 (functional description of the RTC).



Note: V_{PLL2} powers PLL2 and PLL5. V_{PLL3} powers PLL3, PLL4, and PLL6.

Figure 3-2. Clock Generation Block Diagram

General Configuration Block (Continued)

3.5.1 27 MHz Crystal Oscillator

The internal oscillator employs an external crystal connected to the on-chip amplifier. The on-chip amplifier is accessible on the X27I input and X27O output signals. See Figure 3-3 for the recommended external circuit and Table 3-5 for a list of the circuit components.

Choose C_1 and C_2 capacitors to match the crystal's load capacitance. The load capacitance C_L "seen" by crystal Y is comprised of C_1 in series with C_2 and in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout and socket (if any), and can vary from 0 to 10 pF. The rule of thumb in choosing these capacitors is:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PARASITIC}$$

Example 1:

Crystal C_L = 10 pF, $C_{PARASITIC}$ = 8.2 pF
C_1 = 3.6 pF, C_2 = 3.6 pF

Example 2:

Crystal C_L = 20 pF, $C_{PARASITIC}$ = 8 pF
C_1 = 24 pF, C_2 = 24 pF

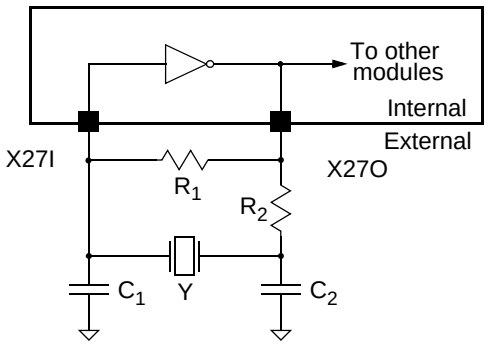


Figure 3-3. Recommended Oscillator External Circuitry

Table 3-5. Crystal Oscillator Circuit Components

Component	Parameters	Values	Tolerance
Crystal	Resonance Frequency	27.00 MHz Parallel mode	50 PPM or better
	Type	AT-cut or BT-cut	
	Serial Resistance	40 Ω	Max
	Shunt Capacitance	7 pF	Max
	Load Capacitance, C_L	10-20 pF	
	Temperature Coefficient	User-defined	
Resistor R_1	Resistance	20 MΩ	5%
Resistor R_2[1]	Resistance	100 Ω	5%
Capacitor C_1[1]	Capacitance	3-24 pF	5%
Capacitor C_2[1]	Capacitance	3-24 pF	5%

1. The value of these components is recommended. It should be tuned according to crystal and board parameters.

General Configuration Block (Continued)

3.5.2 GX1 Module Core Clock

The core clock is generated by an Analog Delay Loop (ADL) clock generator from the internal Fast-PCI clock. The clock can be any whole-number multiple of the input clock between 4 and 10. Possible values are listed in Table 3-6.

At power-on reset, the core clock multiplier value is set according to the value of four strapped balls - CLKSEL[3:0] (EBGA balls AL13, AK3, B27, F3 / TEPBGA balls P30, D29, AF3, B8). These balls also select the clock which is used as input to the multiplier, as shown in Table 3-7.

3.5.3 Internal Fast-PCI Clock

The internal Fast-PCI clock can be configured to 33, 48, or 66 MHz via strap options on the CLKSEL1 and CLKSEL0 balls. These can be read in the internal Fast-PCI Clock field in the CCFC register (GCB+I/O Offset 1Eh[9:8]). (See Table 3-8 on page 95 details on the CCFC register.)

Table 3-6. Core Clock Frequency

ADL Multiplier Value	Internal Fast-PCI Clock Freq. (MHz)		
	33.33	48	66.67
4	133.3	192	266.7
5	166.7	240	---
6	200	288	---
7	233.3	---	---
8	266.7	---	---
9	300	---	---
10	---	---	---

Table 3-7. Strapped Core Clock Frequency

CLKSEL[3:0] Straps	Internal Fast-PCI Clock Freq. (MHz) (GCB+I/O Offset 1Eh[9:8])	Default ADL Multiplier		Maximum Core Clock Freq. (MHz)
		Multiply By	Multiplier Value (GCB+I/O Offset 1Eh[3:0])	
0111	33.33	4	0100	133
1011		5	0101	167
1111		6	0110	200
0000		7	0111	233
0100		8	1000	267
1000		9	1001	300
1100		10	1010	Reserved
0001	48	4	0100	192
0101		5	0101	240
1001		6	0110	288
1101		7	0111	Reserved
0110	66.67	4	0100	267
1010		5	0101	Reserved

General Configuration Block (Continued)

3.5.4 SuperI/O Clocks

The SuperI/O module requires a 48 MHz input for Fast infrared (FIR), UART, and other functions. This clock is supplied by PLL4 using a multiplier value of 576/(108x3) to generate 48 MHz.

3.5.5 Core Logic Module Clocks

The Core Logic module requires the following clock sources:

Real-Time Clock (RTC)

RTC requires a 32.768 KHz clock which is supplied directly from an internal low-power crystal oscillator. This oscillator uses battery power and has very low current consumption.

USB

The USB requires a 48 MHz input which is supplied by PLL4. The required total frequency accuracy and slow jitter for USB is 500 PPM; edge to edge jitter is ±1.2%.

ACPI

The ACPI logic block uses a 14.32 MHz clock supplied by PLL6. PLL6 creates this clock from the 32.768 KHz clock, with a multiplier value of 6992/4 to output a 57.278 MHz clock that is divided by 4.

External PCI

The PCI Interface uses a 33.3 MHz clock that is created by PLL5 and divided by 2. PLL5 uses the 27 MHz clock, to output a 66.67 MHz clock. PLL5 has a frequency accuracy of ± 0.1%.

AC97

The SC2200 generates the 24.576 MHz clock required by the audio codec. Therefore, no crystal need be included for the audio codec on the system board.

PLL3 uses the crystal oscillator clock, to generate a 24.576 MHz clock. This clock is driven on the AC97_CLK ball. The accuracy of the clock supplied by the SC2200 is 50 PPM.

3.5.6 Video Processor Clocks

The Video processor requires the following clock sources:

Dot

The Dot clock is generated by PLL2. It is supplied to the Display Controller in the GX1 module (DCLK) that creates the pixel information, and is returned to the Graphics block (PCLK) with this information. PLL2 uses the 27 MHz clock to generate the Dot clock.

Video

The Video clock source depends on the source of the video data.

- If the video data is coming from the GX1 module (Capture Video mode), the video clock is generated by the Display Controller.
- If the video data is coming directly from the VIP block (Direct Video mode), the Video Clock is generated by the VIP block.

3.5.7 Clock Registers

Table 3-8 describes the registers of the clock generator and PLL.

Table 3-8. Clock Generator Configuration

Bit	Description
Offset 10h **Maximum Core Clock Multiplier Register - MCCM (RO)** **Reset Value: Strapped Value** This register holds the maximum core clock multiplier value. The maximum clock frequency allowed by the core, is the Fast-PCI clock multiplied by this value.	
7:4	**Reserved.**
3:0	**MCM (Maximum Clock Multiplier).** This 4-bit value is the maximum multiplier value allowed for the core clock generator. It is derived from strap pins CLKSEL[3:0] based on the multiplier value in Table 3-7 on page 94.
Offset 11h **Reserved - RSVD**	
Offset 12h **PLL Power Control Register - PPCR (R/W)** **Reset Value: 2Fh** This register controls operation of the PLLs.	
7	**Reserved.**
6	**EXPCID (Disable External PCI Clock).** 0: External PCI clock is enabled. 1: External PCI clock is disabled.
5	**GPD (Disable Graphic Pixel Reference Clock).** 0: PLL2 input clock is enabled. 1: PLL2 input clock is disabled.
4	**Reserved.**

General Configuration Block (Continued)

Table 3-8. Clock Generator Configuration (Continued)

Bit	Description
3	**PLL3SD (Shut Down PLL3).** AC97 codec clock. 0: PLL3 is enabled. 1: PLL3 is shutdown.
2	**FM1SD (Shut Down PLL4).** 0: PLL4 is enabled. 1: PLL4 is shutdown, unless internal Fast-PCI clock is strapped to 48 MHz.
1	**C48MD (Disable SuperI/O and USB Clock).** 0: USB and SuperI/O clock is enabled. 1: USB and SuperI/O clock is disabled.
0	**Reserved.** Write as read.
Offset 13h-17h	**Reserved - RSVD**
Offset 18h-1Bh	**PLL3 Configuration Register - PLL3C (R/W)** **Reset Value: E1040005h**
31:24	**MFFC (MFF Counter Value).** Fvco = OSCCLK * MFBC / (MFFC * MOC) OSCCLK = 27 MHz
23:19	**Reserved.** Write as read.
18:8	**MFBC (MFB Counter Value).** Fvco = OSCCLK * MFBC / (MFFC * MOC) OSCCLK = 27 MHz **Note:** Bits 18, 9, and 8 cannot be changed. Bit 18 is always a 1; bits 9 and 8 are always 0.
7	**Reserved.** Write as read.
6	**Reserved.** Must be set to 0.
5:0	**MOC (MO Counter Value).** Fvco = OSCCLK * MFBC / (MFFC * MOC) OSCCLK = 27 MHz
Offset 1Eh-1Fh	**Core Clock Frequency Control Register - CCFC (R/W)** **Reset Value: Strapped Value** This register controls the configuration of the core clock multiplier and the reference clocks.
15:14	**Reserved.**
13	**Reserved.** Must be set to 0.
12	**Reserved.** Must be set to 0.
11:10	**Reserved.**
9:8	**FPCICK (Internal Fast-PCI Clock). (Read Only)** Reflects the internal Fast-PCI clock and is the input to the GX1 module that is used to generate the core clock. These bits reflect the value of strap pins CLKSEL[1:0]. 00: 33.3 MHz 01: 48 MHz 10: 66.7 MHz 11: 33.3 MHz
7:4	**Reserved.**

General Configuration Block (Continued)

Table 3-8. Clock Generator Configuration (Continued)

Bit	Description
3:0	**MVAL (Multiplier Value).** This 4-bit value controls the multiplier in ADL. The value is set according to the Maximum Clock Multiplier bits of the MCCM register (Offset 10h). The multiplier value should never be written with a multiplier which is different from the multiplier indicated in the MCCM register. 0100: Multiply by 4 0101: Multiply by 5 0110: Multiply by 6 0111: Multiply by 7 1000: Multiply by 8 1001: Multiply by 9 1010: Multiply by 10 Other: Reserved

4.0 SuperI/O Module

The SuperI/O (SIO) module is a member of National Semiconductor's SuperI/O family of integrated PC peripherals. It is a PC98 and ACPI compliant SIO that offers a single-cell solution to the most commonly used ISA peripherals.

The SIO module incorporates: two Serial Ports, an Infrared Communication Port that supports FIR, MIR, HP-SIR, Sharp-IR, and Consumer Electronics-IR, a full IEEE 1284 Parallel Port, two ACCESS.bus Interface (ACB) ports, System Wakeup Control (SWC), and a Real-Time Clock (RTC) that provides RTC timekeeping.

Outstanding Features

- Full compatibility with ACPI Revision 1.0 requirements.
- System Wakeup Control powered by V_{SB}, generates power-up request and a PME (power management event) in response to SDATA_IN2 (an audio codec), IRRX1 (a pre-programmed CEIR), or a RI2# (Serial Port ring indicate) event.
- Advanced RTC, Y2K compliant.

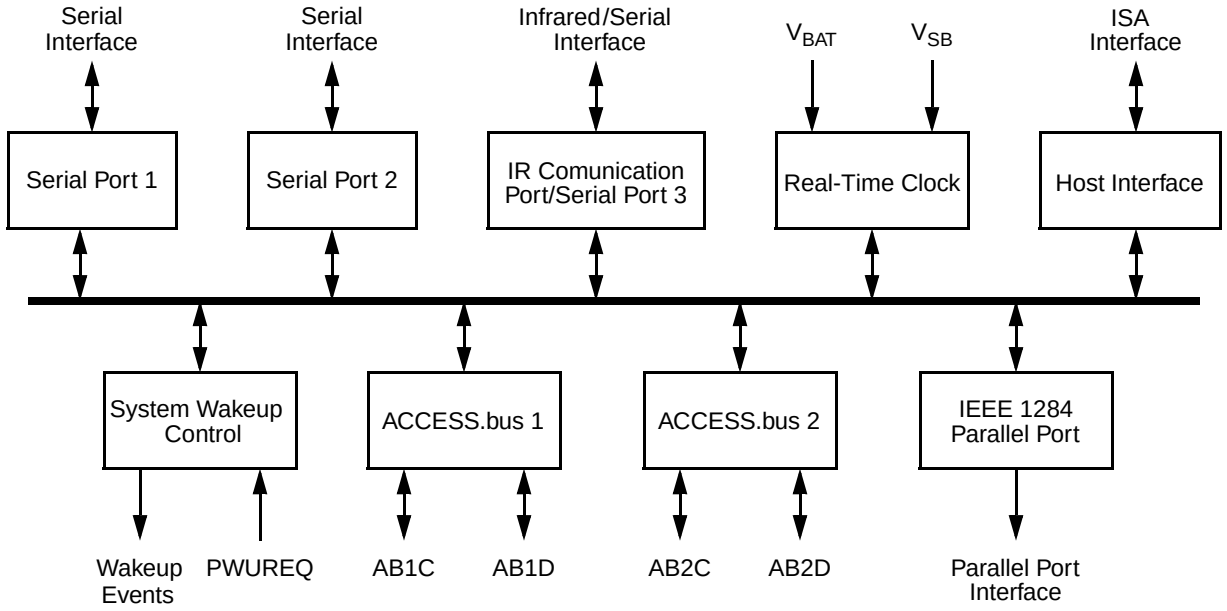


Figure 4-1. SIO Block Diagram

SuperI/O Module (Continued)

4.1 FEATURES

PC98 and ACPI Compliant

- PnP Configuration Register structure
- Flexible resource allocation for all logical devices:
 - — Relocatable base address
 - — 9 parallel IRQ routing options
 - — 3 optional 8-bit DMA channels (where applicable)

Parallel Port

- Software or hardware control
- Enhanced Parallel Port (EPP) compatible with version EPP 1.9 and IEEE 1284 compliant
- EPP support for version EPP 1.7 of the Xircom specification
- EPP support as mode 4 of the Extended Capabilities Port (ECP)
- IEEE 1284 compliant ECP, including level 2
- Selection of internal pull-up or pull-down resistor for Paper End (PE) pin
- PCI bus utilization reduction by supporting a demand DMA mode mechanism and a DMA fairness mechanism
- Protection circuit that prevents damage to the Parallel Port when a printer connected to it powers up or is operated at high voltages, even if the device is in power-down
- Output buffers that can sink and source 14 mA

Serial Port 1

- 16550A compatible (SIN1, SOUT1, DTR1#/BOUT1 signals only)

Serial Port 2

- 16550A compatible

Serial Port 3 / Infrared (IR) Communication Port

- Serial Port 3
 - — SIN and SOUT signals only
 - — Data rate of up to 1.5 Mbps
 - — Software compatible with the 16550A and the 16450
 - — Shadow register support for write-only bit monitoring
 - — DMA support
- IR Communication Port
 - — IrDA 1.1 and 1.0 compatible
 - — Data rate of up to 115.2 Kbps (HP-SIR)
 - — Data rate of 1.152 Mbps (MIR)
 - — Data rate of 4.0 Mbps (FIR)
 - — Selectable internal or external modulation/demodulation (ASK-IR and DASK-IR options of SHARP-IR)
 - — Consumer-IR (TV-Remote) mode
 - — Consumer Remote Control supports RC-5, RC-6, NEC, RCA and RECS 80
 - — DMA support

System Wakeup Control (SWC)

- Power-up request upon detection of RI2#, CEIR, or SDATA_IN2 activity:
 - — Optional routing of power-up request on IRQ line
- Pre-programmed CEIR address in a pre-selected standard (any NEC, RCA or RC-5)
- Powered by V_{SB}
- Battery-backed wakeup setup
- Power-fail recovery support

Real-Time Clock

- A modifiable address that is referenced by a 16-bit programmable register
- DS1287, MC146818 and PC87911 compatibility
- 242 bytes of battery backed up CMOS RAM in two banks
- Selective lock mechanisms for the CMOS RAM
- Battery backed up century calendar in days, day of the week, date of month, months, years and century, with automatic leap-year adjustment
- Battery backed-up time of day in seconds, minutes and hours that allows a 12 or 24 hour format and adjustments for daylight savings time
- BCD or binary format for time keeping
- Three different maskable interrupt flags:
 - — Periodic interrupts - At intervals from 122 msec to 500 msec
 - — Time-of-Month alarm - At intervals from once per second to once per month
 - — Update Ended Interrupt - Once per second upon completion of update
- Separate battery pin, 3.0V operation that includes an internal UL protection resistor
- 7 µA typical power consumption during power down
- Double-buffer time registers
- Y2K Compliant

Clock Sources

- 48 MHz clock input
- On-chip low frequency clock generator for wakeup
- 32.768 KHz crystal with an internal frequency multiplier to generate all required internal frequencies

SuperI/O Module (Continued)

4.2 MODULE ARCHITECTURE

The SIO module comprises a collection of generic functional blocks. Each functional block is described in detail later in this chapter. The beginning of this chapter describes the SIO structure and provides all device specific information, including special implementation of generic blocks, system interface and device configuration.

The SIO module is based on eight logical devices, the host interface, and a central configuration register set, all built around a central, internal 8-bit bus.

The host interface serves as a bridge between the external ISA interface and the internal bus. It supports 8-bit I/O read, 8-bit I/O write and 8-bit DMA transactions, as defined in *Personal Computer Bus Standard P996*.

The central configuration register set supports ACPI compliant PnP configuration. The configuration registers are structured as a subset of the Plug and Play Standard Registers, defined in Appendix A of the *Plug and Play ISA Specification* Version 1.0a by Intel and Microsoft. All system resources assigned to the functional blocks (I/O address space, DMA channels and IRQ lines) are configured in, and managed by, the central configuration register set. In addition, some function-specific parameters are configurable through this unit and distributed to the functional blocks through special control signals.

The source of the device internal clocks is the 48 MHz clock signal or through the 32.768 KHz crystal with an internal frequency multiplier. RTC operates on a 32 KHz clock.

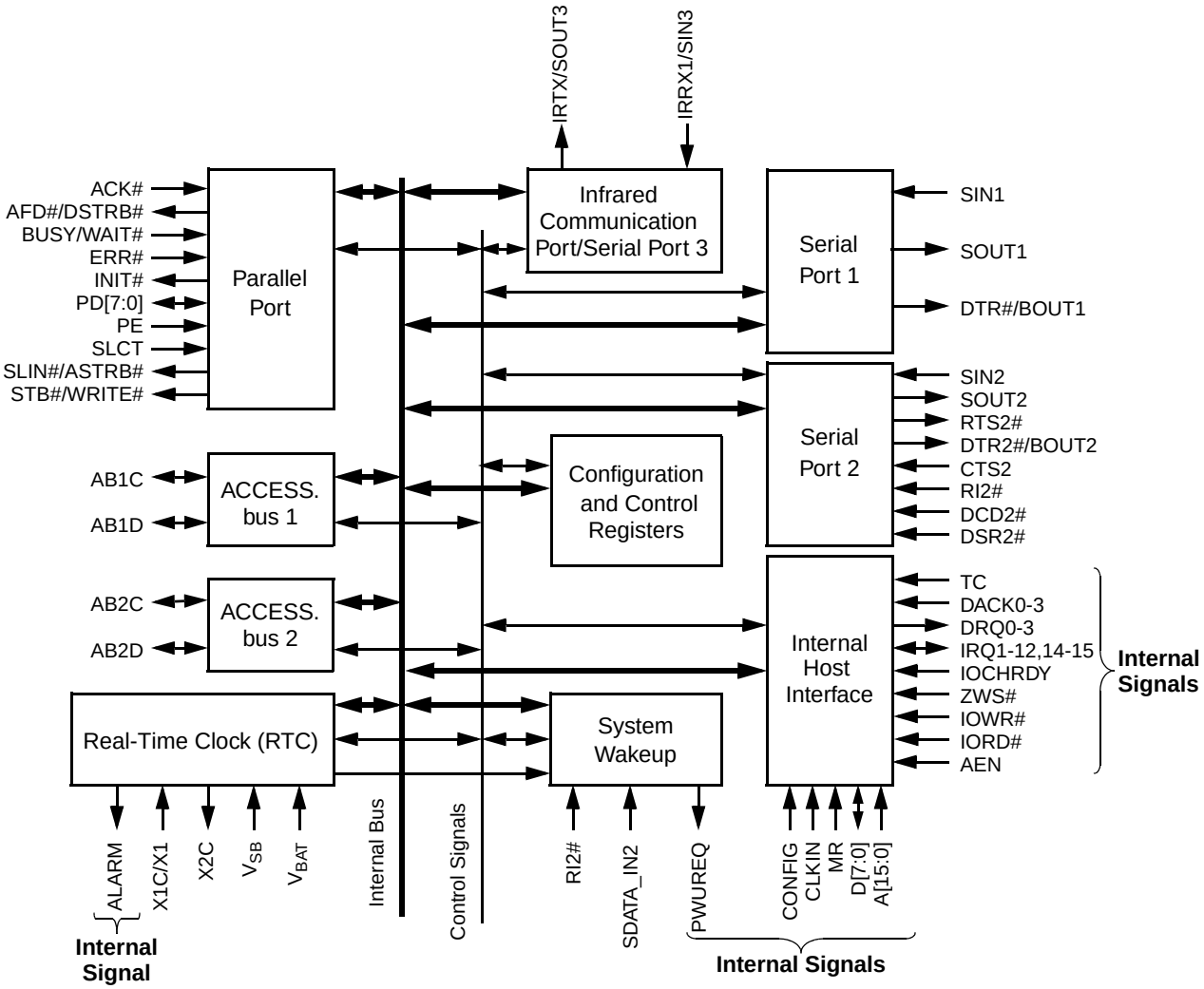


Figure 4-2. Detailed SIO Block Diagram

4.3 CONFIGURATION STRUCTURE/ACCESS

This section describes the structure of the configuration register file, and the method of accessing the configuration registers.

4.3.1 Index-Data Register Pair

The SIO configuration access is performed via an Index-Data register pair, using only two system I/O byte locations. The base address of this register pair is determined according to the state of the IO_SIOCFG_IN bit field of the Core Logic module (F5BAR0+I/O Offset 00h[26:25]). Table 4-1 shows the selected base addresses as a function of the IO_SIOCFG_IN bit field.

Table 4-1. SIO Configuration Options

IO_SIOCFG_IN Settings	I/O Address		Description
	Index Register	Data Register	
00	-	-	SIO disabled
01	-	-	Configuration access disabled
10	002Eh	002Fh	Base address 1 selected
11	015Ch	015Dh	Base address 2 selected

The Index Register is an 8-bit R/W register located at the selected base address (Base+0). It is used as a pointer to the configuration register file, and holds the index of the configuration register that is currently accessible via the Data Register. Reading the Index Register returns the last value written to it (or the default of 00h after reset).

The Data Register is an 8-bit virtual register, used as a data path to any configuration register. Accessing the data register results with physically accessing the configuration register that is currently pointed by the Index Register.

4.3.2 Banked Logical Device Registers

Each functional block is associated with a Logical Device Number (LDN). The configuration registers are grouped into banks, where each bank holds the standard configuration registers of the corresponding logical device. Table 4-2 shows the LDNs of the device functional blocks.

Table 4-2. LDN Assignments

LDN	Functional Block	Reference
00h	Real-Time Clock (RTC)	Page 107
01h	System Wakeup Control (SWC)	Page 109
02h	Infrared Communication Port (IRCP) or Serial Port 3 (SP3)	Page 110
03h	Serial Port 1 (SP1)	Page 111
05h	ACCESS.bus 1 (ACB1)	Page 112
06h	ACCESS.bus 2 (ACB2)	
07h	Parallel Port (PP)	Page 113
08h	Serial Port 2 (SP2)	Page 111

Figure 4-3 shows the structure of the standard PnP configuration register file. The SIO Control And Configuration registers are not banked and are accessed by the Index-Data register pair only (as described above). However, the Logical Device Control and Configuration registers are duplicated over eight banks for eight logical devices. Therefore, accessing a specific register in a specific bank is performed by two-dimensional indexing, where the LDN register selects the bank (or logical device), and the Index register selects the register within the bank. Accessing the Data register while the Index register holds a value of 30h or higher results in a physical access to the Logical Device Configuration registers currently pointed to by the Index register, within the logical device bank currently selected by the LDN register.

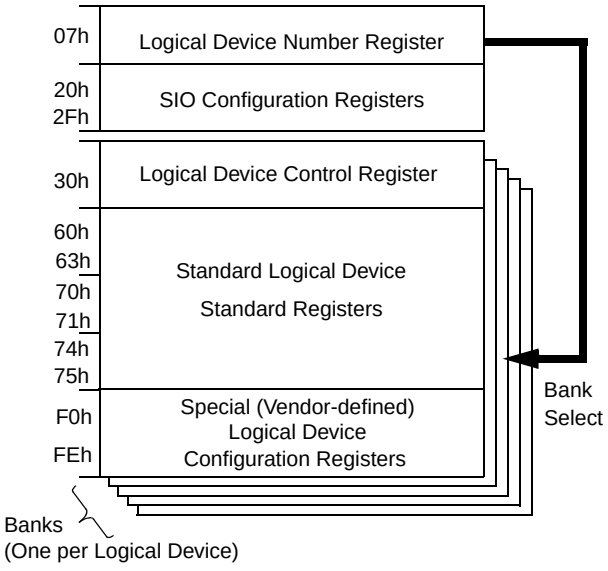


Figure 4-3. Structure of the Standard Configuration Register File

SuperI/O Module (Continued)

Write accesses to unimplemented registers (i.e., accessing the Data register while the Index register points to a non-existing register or the LDN is 07h or higher than 08h), are ignored and a read returns 00h on all addresses except for 74h and 75h (DMA configuration registers) which returns 04h (indicating no DMA channel is active). The configuration registers are accessible immediately after reset.

4.3.3 Default Configuration Setup

The device has four reset types:

Software Reset

This reset is generated by bit 1 of the SIOCF1 register, which resets all logical devices. A software reset also resets most bits in the SIO Configuration and Control registers (see Section 4.4.1 on page 106 for the bits not affected). This reset does not affect register bits that are locked for write access.

Hardware Reset

This reset is activated by the system reset signal. This resets all logical devices, with the exception of the RTC and the SWC, and all SIO Configuration and Control registers, with the exception of the SIOCF2 register. It also resets all SuperI/O control and configuration registers, except for those that are battery-backed.

V_{PP} Power-Up Reset

This reset is activated when either V_{SB} or V_{BAT} is powered on after both have been off. V_{PP} is an internal voltage which is a combination of V_{SB} and V_{BAT}. V_{PP} is taken from V_{SB} if V_{SB} is greater than the minimum (Min) value defined in Section 8.1.3 "Operating Conditions" on page 358; otherwise, V_{BAT} is used as the V_{PP} source. This reset resets all registers whose values are retained by V_{PP}.

V_{SB} Power-Up Reset

This is an internally generated reset that resets the SWC, excluding those SWC registers whose values are retained by V_{PP}. This reset is activated after V_{SB} is powered up.

The SIO module wakes up with the default setup, as follows:

- When a hardware reset occurs:
 - The configuration base address is 2Eh, 15Ch or None, according to the IO_SIOCFG_IN bit values, as shown in Table 4-1 on page 101.
 - All Logical devices are disabled, with the exception of the RTC and the SWC, which remains functional but whose registers cannot be accessed.
- When either a hardware or a software reset occurs:
 - The legacy devices are assigned with their legacy system resource allocation.
 - The National proprietary functions are not assigned with any default resources and the default values of their base addresses are all 00h.

4.3.4 Address Decoding

A full 16-bit address decoding is applied when accessing the configuration I/O space, as well as the registers of the functional blocks. However, the number of configurable bits in the base address registers vary for each device.

The lower 1, 2, 3 or 4 address bits are decoded within the functional block to determine the offset of the accessed register, within the device's I/O range of 2, 4, 8 or 16 bytes, respectively. The rest of the bits are matched with the base address register to decode the entire I/O range allocated to the device. Therefore the lower bits of the base address register are forced to 0 (RO), and the base address is forced to be 2, 4, 8 or 16 byte aligned, according to the size of the I/O range.

The base address of the RTC, Serial Port 1, Serial Port 2, and the Infrared Communication Port are limited to the I/O address range of 00h to 7Fxh only (bits [15:11] are forced to 0). The Parallel Port base address is limited to the I/O address range of 00h to 3F8h. The addresses of the non-legacy devices are configurable within the full 16-bit address range (up to FFFxh).

In some special cases, other address bits are used for internal decoding (such as 10 in the Parallel Port). For more details, please see the detailed description of the base address register for each specific logical device.

4.4 STANDARD CONFIGURATION REGISTERS

As illustrated in Figure 4-4, the Standard Configuration registers are broadly divided into two categories: SIO Control and Configuration registers and Logical Device Control and Configuration registers (one per logical device, some are optional).

SIO Control and Configuration Registers

The only PnP control register in the SIO module is the Logical Device Number register at Index 07h. All other standard PnP control registers are associated with PnP protocol for ISA add-in cards, and are not supported by the SIO module.

The SIO Configuration registers at Index 20h-27h are mainly used for part identification. (See Section 4.4.1 "SIO Control and Configuration Registers" on page 106 for further details.)

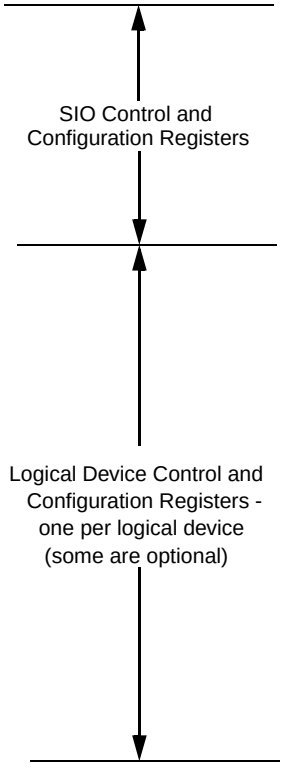
Logical Device Control and Configuration Registers

A subset of these registers is implemented for each logical device. (See Table 4-2 on page 101 for LDN assignment and Section 4.4.2 "Logical Device Control and Configuration" on page 107 for register details.)

Logical Device Control Register (Index 30h): The only implemented Logical Device Control register is the Activate register at Index 30. Bit 0 of the Activate register and bit 0 of the SIO Configuration 1 register (Global Device Enable bit) control the activation of the associated function block (except for the RTC and the SWC). Activation of the block enables access to the block's registers, and attaches its system resources, which are unused as long as the block is not activated. Activation of the block may also result in other effects (e.g., clock enable and active signaling), for certain functions.

Standard Logical Device Configuration Registers (Index 60h-75h): These registers are used to manage the resource allocation to the functional blocks. The I/O port base address descriptor 0 is a pair of registers at Index 60h-61h, holding the (first or only) 16-bit base address for the register set of the functional block. An optional second base-address (descriptor 1) at Index 62h-63h is used for devices with more than one continuous register set. Interrupt Number Select (Index 70h) and Interrupt Type Select (Index 71h) allocate an IRQ line to the block and control its type. DMA Channel Select 0 (Index 74h) allocates a DMA channel to the block, where applicable. DMA Channel Select 1 (Index 75h) allocates a second DMA channel, where applicable.

Special Logical Device Configuration Registers (F0h-F3h): The vendor-defined registers, starting at Index F0h are used to control function-specific parameters such as operation modes, power saving modes, pin TRI-STATE, clock rate selection, and non-standard extensions to generic functions.



Index	Register Name
07h	Logical Device Number
20h	SIO ID
21h	SIO Configuration 1
22h	SIO Configuration 2
27h	SIO Revision ID
2Eh	Reserved exclusively for National use
30h	Logical Device Control (Activate)
60h	I/O Port Base Address Descriptor 0 Bits [15:8]
61h	I/O Port Base Address Descriptor 0 Bits [7:0]
62h	I/O Port Base Address Descriptor 1 Bits [15:8]
63h	I/O Port Base Address Descriptor 1 Bits [7:0]
70h	Interrupt Number Select
71h	Interrupt Type Select
74h	DMA Channel Select 0
75h	DMA Channel Select 1
F0h	Device Specific Logical Device Configuration 1
F1h	Device Specific Logical Device Configuration 2
F2h	Device Specific Logical Device Configuration 3
F3h	Device Specific Logical Device Configuration 4

Figure 4-4. Standard Configuration Registers Map

SuperI/O Module (Continued)

Table 4-3 provides the bit definitions for the Standard Configuration registers.

- All reserved bits return 0 on reads, except where noted otherwise. They must not be modified as such modification may cause unpredictable results. Use read-modify-write to prevent the values of reserved bits from being changed during write.
- Write only registers should not use read-modify-write during updates.

Table 4-3. Standard Configuration Registers

Bit	Description
Index 07h	**Logical Device Number (R/W)**
This register selects the current logical device. See Table 4-2 for valid numbers. All other values are reserved.	
7:0	**Logical Device number.**
Index 20h-2Fh	**SIO Configuration (R/W)**
SIO configuration and ID registers. See Section 4.4.1 "SIO Control and Configuration Registers" on page 106 for register/bit details.	
Index 30h	**Activate (R/W)**
7:1	**Reserved.**
0	**Logical Device Activation Control.** 0: Disable. 1: Enable.
Index 60h	**I/O Port Base Address Bits [15:8] Descriptor 0 (R/W)**
7:0	**Descriptor 0 A[15:8].** Selects I/O lower limit address bits [15:8] for I/O Descriptor 0.
Index 61h	**I/O Port Base Address Bits [7:0] Descriptor 0 (R/W)**
7:0	**Descriptor 0 A[7:0].** Selects I/O lower limit address bits [7:0] for I/O Descriptor 0.
Index 62h	**I/O Port Base Address Bits [15:8] Descriptor 1 (R/W)**
7:0	**Descriptor 1 A[15:8].** Selects I/O lower limit address bits [15:8] for I/O Descriptor 1.
Index 63h	**I/O Port Base Address Bits [7:0] Descriptor 1 (R/W)**
7:0	**Descriptor 1 A[7:0].** Selects I/O lower limit address bits [7:0] for I/O Descriptor 1.
Index 70h	**Interrupt Number (R/W)**
7:4	**Reserved.**
3:0	**Interrupt Number.** These bits select the interrupt number. A value of 1 selects IRQ1, a value of 2 selects IRQ2, etc. (up to IRQ12). **Note:** IRQ0 is not a valid interrupt selection.
Index 71h	**Interrupt Request Type Select (R/W)**
Selects the type and level of the interrupt request number selected in the previous register.	
7:2	**Reserved.**
1	**Interrupt Level Requested.** Level of interrupt request selected in previous register. 0: Low polarity. 1: High polarity. This bit must be set to 1 (high polarity), except for IRQ8#, that must be low polarity.
0	**Interrupt Type Requested.** Type of interrupt request selected in previous register. 0: Edge. 1: Level.
Index 74h	**DMA Channel Select 0 (R/W)**
Selects selected DMA channel for DMA 0 of the logical device (0 - the first DMA channel in case of using more than one DMA channel).	
7:3	**Reserved.**
2:0	**DMA 0 Channel Select.** This bit field selects the DMA channel for DMA 0. The valid choices are 0-3, where a value of 0 selects DMA channel 0, 1 selects channel 1, etc. A value of 4 indicates that no DMA channel is active. Values 5-7 are reserved.

SuperI/O Module (Continued)

Table 4-3. Standard Configuration Registers

Bit	Description
Index 75h	**DMA Channel Select 1 (R/W)**
Indicates selected DMA channel for DMA 1 of the logical device (1 - the second DMA channel in case of using more than one DMA channel).	
7:3	**Reserved.**
2:0	**DMA 1 Channel Select:** This bit field selects the DMA channel for DMA 1. The valid choices are 0-3, where a value of 0 selects DMA channel 0, 1 selects channel 1, etc. A value of 4 indicates that no DMA channel is active. Values 5-7 are reserved.
Index F0h-FEh	**Logical Device Configuration (R/W)**
Special (vendor-defined) configuration options.	

SuperI/O Module (Continued)

4.4.1 SIO Control and Configuration Registers

Table 4-4 lists the SIO Control and Configuration registers and Table 4-5 provides their bit formats.

Table 4-4. SIO Control and Configuration Register Map

Index	Type	Name	Power Rail	Reset Value
20h	RO	**SID.** SIO ID	V_{CORE}	F5h
21h	R/W	**SIOCF1.** SIO Configuration 1	V_{CORE}	01h
22h	R/W	**SIOCF2.** SIO Configuration 2	V_{PP}	02h
27h	RO	**SRID.** SIO Revision ID	V_{CORE}	01h
2Eh	---	**RSVD.** Reserved exclusively for National use	---	---

Table 4-5. SIO Control and Configuration Registers

Bit	Description
Index 20h	**SIO ID Register - SID (RO)** **Reset Value: F5h**
7:0	**Chip ID.** Contains the identity number of the module. The SIO module is identified by the value F5h.
Index 21h	**SIO Configuration 1 Register - SIOCF1 (RW)** **Reset Value: 01h**
7:6	**General Purpose Scratch.** When bit 5 is set to 1, these bits are RO. After reset, these bits can be read or write. Once changed to RO, the bits can be changed back to R/W only by a hardware reset.
5	**Lock Scratch.** This bit controls bits 7 and 6 of this register. Once this bit is set to 1 by software, it can be cleared to 0 only by a hardware reset. 0: Bits 7 and 6 of this register are R/W bits. (Default) 1: Bits 7 and 6 of this register are RO bits.
4:2	**Reserved.**
1	**SW Reset**. Read always returns 0. 0: Ignored. (Default) 1: Resets all devices that are reset by MR (with the exception of the lock bits) and the registers of the SWC.
0	**Global Device Enable.** This bit controls the function enable of all the logical devices in the SIO module, except the SWC and the RTC. It allows them to be disabled simultaneously by writing to a single bit. 0: All logical devices in the SIO module are disabled, except the SWC and the RTC. 1: Each logical device is enabled according to its Activate register at Index 30h. (Default)
Index 22h	**SIO Configuration 2 Register - SIOCF2 (R/W)** **Reset Value: 02h**
Note:	This register is reset only when V_{PP} is first applied.
7	**Reserved.**
6:4	**General Purpose Scratch.** Battery-backed.
3:2	**Reserved.**
1	**Reserved.**
0	**Reserved. (RO)**
Index 27h	**SIO Revision ID Register - SRID (RO)** **Reset Value: 01h**
7:0	**SIO Revision ID. (RO)** This RO register contains the identity number of the chip revision. SRID is incremented on each revision.

SuperI/O Module (Continued)

4.4.2 Logical Device Control and Configuration

As described in Section 4.3.2 "Banked Logical Device Registers" on page 101, each functional block is associated with a Logical Device Number (LDN). This section provides the register descriptions for each LDN.

The register descriptions in this subsection use the following abbreviations for Type:

- R/W = Read/Write
- R = Read from a specific address returns the value of a specific register. Write to the same address is to a different register.
- W = Write
- RO = Read Only
- R/W1C = Read/Write 1 to Clear. Writing 1 to a bit clears it to 0. Writing 0 has no effect.

4.4.2.1 LDN 00h - Real-Time Clock

Table 4-6 lists the registers which are relevant to configuration of the Real-Time Clock (RTC). Only the last registers (F0h-F3h) are described here (Table 4-7). See Table 4-3 "Standard Configuration Registers" on page 104 for descriptions of the other registers.

Table 4-6. Relevant RTC Configuration Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** When bit 0 is cleared, the registers of this logical device are not accessible.[1]	00h
60h	R/W	**Standard Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	00h
61h	R/W	**Standard Base Address LSB register.** Bit 0 (for A0) is RO, 0b.	70h
62h	R/W	**Extended Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	00h
63h	R/W	**Extended Base Address LSB register.** Bit 0 (for A0) is RO, 0b.	72h
70h	R/W	**Interrupt Number.**	08h
71h	R/W	**Interrupt Type.** Bit 1 is R/W; other bits are RO.	00h
74h	RO	**Report no DMA assignment.**	04h
75h	RO	**Report no DMA assignment.**	04h
F0h	R/W	**RAM Lock register (RLR).**	00h
F1h	R/W	**Date of Month Alarm Offset register (DOMAO).** Sets index of Date of Month Alarm register in the standard base address.	00h
F2h	R/W	**Month Alarm Offset register (MONAO).** Sets index of Month Alarm register in the standard base address.	00h
F3h	R/W	**Century Offset register (CENO).** Sets index of Century register in the standard base address.	00h

1. The logical device registers are maintained, and all RTC mechanisms are functional.

SuperI/O Module (Continued)

Table 4-7. RTC Configuration Registers

Bit	Description
Index F0h	**RAM Lock Register - RLR (R/W)**
When any non-reserved bit in this register is set to 1, it can be cleared only by hardware reset.	
7	**Block Standard RAM.** 0: No effect on Standard RAM access. (Default) 1: Read and write to locations 38h-3Fh of the Standard RAM are blocked, writes ignored, and reads return FFh.
6	**Block RAM Write.** 0: No effect on RAM access. (Default) 1: Writes to RAM (Standard and Extended) are ignored.
5	**Block Extended RAM Write.** This bit controls writes to bytes 00h-1Fh of the Extended RAM. 0: No effect on the Extended RAM access. (Default) 1: Writes to bytes 00h-1Fh of the Extended RAM are ignored.
4	**Block Extended RAM Read.** This bit controls read from bytes 00h-1Fh of the Extended RAM. 0: No effect on Extended RAM access. (Default) 1: Reads to bytes 00h-1Fh of the Extended RAM are ignored.
3	**Block Extended RAM.** This bit controls access to the Extended RAM 128 bytes. 0: No effect on Extended RAM access. (Default) 1: Read and write to the Extended RAM are blocked: writes are ignored and reads return FFh.
2:0	**Reserved.**
Index F1h	**Date Of Month Alarm Register Offset Register - DOMAO (R/W)**
7	**Reserved.**
6:0	**Date of Month Alarm Register Offset Value.**
Index F2h	**Month Alarm Register Offset Register - MANAO (R/W)**
7	**Reserved.**
6:0	**Month Alarm Register Offset Value.**
Index F3h	**Century Register Offset Register - CENO (R/W)**
7	**Reserved.**
6:0	**Century Register Offset Value.**

SuperI/O Module (Continued)

4.4.2.2 LDN 01h - System Wakeup Control

Table 4-8 lists registers that are relevant to the configuration of System Wakeup Control (SWC). These registers are described earlier in Table 4-3 "Standard Configuration Registers" on page 104.

Table 4-8. Relevant SWC Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** When bit 0 is cleared, the registers of this logical device are not accessible.[1]	00h
60h	R/W	**Base Address MSB register.**	00h
61h	R/W	**Base Address LSB register.** Bits [3:0] (for A[3:0]) are RO, 0000b.	00h
70h	R/W	**Interrupt Number.** (For routing the internal PWUREQ signal.)	00h
71h	R/W	**Interrupt Type.** Bit 1 is R/W. Other bits are RO.	03h
74h	RO	**Report no DMA assignment.**	04h
75h	RO	**Report no DMA assignment.**	04h

1. The logical device registers are maintained, and all wakeup detection mechanisms are functional.

SuperI/O Module (Continued)

4.4.2.3 LDN 02h - Infrared Communication Port or Serial Port 3

Table 4-9 lists the configuration registers which affect the Infrared Communication Port or Serial Port 3 (IRCP/SP3). Only the last register (F0h) is described here (Table 4-10). See Table 4-3 "Standard Configuration Registers" on page 104 for descriptions of the other registers listed.

Table 4-9. Relevant IRCP/SP3 Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register.	00h
60h	R/W	**Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	03h
61h	R/W	**Base Address LSB register.** Bit [2:0] (for A[2:0]) are RO, 000b.	E8h
70h	R/W	**Interrupt Number.**	00h
71h	R/W	**Interrupt Type.** Bit 1 is R/W; other bits are RO.	03h
74h	R/W	**DMA Channel Select 0 (RX_DMA)**.	04h
75h	R/W	**DMA Channel Select 1 (TX_DMA).**	04h
F0h	R/W	**Infrared Communication Port/Serial Port 3 Configuration register.**	02h

Table 4-10. IRCP/SP3 Configuration Register

Bit	Description
Index F0h	**Infrared Communication Port/Serial Port 3 Configuration Register (R/W)** **Reset Value: 02h**
7	**Bank Select Enable.** Enables bank switching. 0: All attempts to access the extended registers are ignored. (Default) 1: Enables bank switching.
6:3	**Reserved.**
2	**Busy Indicator.** (RO) This bit can be used by power management software to decide when to power-down the device. 0: No transfer in progress. (Default) 1: Transfer in progress.
1	**Power Mode Control.** When the logical device is active in: 0: Low power mode - Clock disabled. The output signals are set to their default states. Registers are maintained. (Unlike Active bit in Index 30h that also prevents access to device registers.) 1: Normal power mode - Clock enabled. The device is functional when the logical device is active. (Default)
0	**TRI-STATE Control**. When enabled and the device is inactive, the logical device output pins are in TRI-STATE. One exception is the IRTX/SOUT3 pin, which is driven to 0 when the Infrared Communication Port or Serial Port 3 is inactive and is not affected by this bit. 0: Disabled. (Default) 1: Enabled (when the device is inactive).

SuperI/O Module (Continued)

4.4.2.4 LDN 03h and 08h - Serial Ports 1 and 2

Serial Ports 1 and 2 are identical, except for their reset values.

Serial Port 1 is designated as LDN 03h and Serial Port 2 as LDN 08h. Table 4-11 lists the configuration registers which affect Serial Ports 1 and 2. Only the last register (F0h) is described here (Table 4-12). See Table 4-3 "Standard Configuration Registers" on page 104 for descriptions of the others.

Table 4-11. Relevant Serial Ports 1 and 2 Registers

Index	Type	Configuration Register or Action	Reset Value	
			Port 1	Port 2
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register.	00h	00h
60h	R/W	**Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	03h	02h
61h	R/W	**Base Address LSB register**. Bit [2:0] (for A[2:0]) are RO, 000b.	F8h	F8h
70h	R/W	**Interrupt Number.**	04h	03h
71h	R/W	**Interrupt Type.** Bit 1 is R/W; other bits are RO.	03h	03h
74h	RO	**Report no DMA assignment**.	04h	04h
75h	RO	**Report no DMA assignment.**	04h	04h
F0h	R/W	**Serial Ports 1 and 2 Configuration register.**	02h	02h

Table 4-12. Serial Ports 1 and 2 Configuration Register

Bit	Description
Index F0h	**Serial Ports 1 and 2 Configuration Register (R/W)** — **Reset Value: 02h**
7	**Bank Select Enable**. Enables bank switching for Serial Ports 1 and 2. 0: Disabled. (Default) 1: Enabled.
6:3	**Reserved.**
2	**Busy Indicator. (RO)** This bit can be used by power management software to decide when to power-down Serial Ports 1 and 2 logical devices. 0: No transfer in progress. (Default) 1: Transfer in progress.
1	**Power Mode Control.** When the logical device is active in: 0: Low power mode - Serial Ports 1 and 2 Clock disabled. The output signals are set to their default states. Registers are maintained. (Unlike Active bit in Index 30h that also prevents access to Serial Ports 1 or 2 registers.) 1: Normal power mode - Serial Ports 1 and 2 clock enabled. Serial Ports 1 and 2 are functional when the respective logical devices are active. (Default)
0	**TRI-STATE Control**. This bit controls the TRI-STATE status of the device output pins when it is inactive (disabled). 0: Disabled. (Default) 1: Enabled when device inactive.

SuperI/O Module (Continued)

4.4.2.5 LDN 05h and 06h - ACCESS.bus Ports 1 and 2

ACCESS.bus ports 1 and 2 (ACB1 and ACB2) are identical. Each ACB is a two-wire synchronous serial interface compatible with the ACCESS.bus physical layer. ACB1 and ACB2 use a 24 MHz internal clock. Six runtime registers for each ACCESS.bus are described in Section 4.7 "ACCESS.bus Interface" on page 130.

ACB1 is designated as LDN 05h and ACB2 as LDN 06h. Table 4-13 lists the configuration registers which affect the ACCESS.bus ports. Only the last register (F0h) is described here (Table 4-14). See Table 4-3 "Standard Configuration Registers" on page 104 for descriptions of the others.

Table 4-13. Relevant ACB1 and ACB2 Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register	00h
60h	R/W	**Base Address MSB register.**	00h
61h	R/W	**Base Address LSB register.** Bits [2:0] (for A[2:0]) are RO, 000b.	00h
70h	R/W	**Interrupt Number.**	00h
71h	R/W	**Interrupt Type.** Bit 1 is R/W. Other bits are RO.	03h
74h	RO	**Report no DMA assignment.**	04h
75h	RO	**Report no DMA assignment.**	04h
F0h	R/W	**ACB1 and ACB2 Configuration register.**	00h

Table 4-14. ACB1 and ACB2 Configuration Register

Bit	Description
Index F0h **ACB1 and ACB2 Configuration Register (R/W)** This register is reset by hardware to 00h.	
7:3	**Reserved.**
2	**Internal Pull-Up Enable.** 0: No internal pull-up resistors on AB1C/AB2C and AB1D/AB2D. (Default) 1: Internal pull-up resistors on AB1C/AB2C and AB1D/AB2D.
1:0	**Reserved.**

SuperI/O Module (Continued)

4.4.2.6 LDN 07h - Parallel Port

The Parallel Port supports all IEEE 1284 standard communication modes: Compatibility (known also as Standard or SPP), Bidirectional (known also as PS/2), FIFO, EPP (known also as Mode 4) and ECP (with an optional Extended ECP mode).

The Parallel Port includes two groups of runtime registers, as follows:

- A group of 21 registers at first level offset, sharing 14 entries. Three of these registers (at Offset 403h, 404h, and 405h) are used only in the Extended ECP mode.
- A group of four registers, used only in the Extended ECP mode, accessed by a second level offset.

The desired mode is selected by the ECR runtime register (Offset 402h). The selected mode determines which runtime registers are used and which address bits are used for the base address. (See Section 4.8.1 on page 138 for further details regarding the runtime registers.)

Table 4-15 lists the configuration registers which affect the Parallel Port. Only the last register (F0h) is described here (Table 4-16). See Table 4-3 "Standard Configuration Registers" on page 104 for descriptions of the others.

Table 4-15. Relevant Parallel Port Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register.	00h
60h	R/W	**Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b. Bit 2 (for A10) should be 0b.	02h
61h	R/W	**Base Address LSB register.** Bits 1 and 0 (A1 and A0) are RO, 00b. For ECP Mode 4 (EPP) or when using the Extended registers, bit 2 (A2) should also be 0b.	78h
70h	R/W	**Interrupt Number.**	07h
71h	R/W	**Interrupt Type**. Bits [7:2] are RO. Bit 1 is R/W. Bit 0 is RO. It reflects the interrupt type dictated by the Parallel Port operation mode. This bit is set to 1 (level interrupt) in Extended Mode and cleared (edge interrupt) in all other modes.	02h
74h	R/W	**DMA Channel Select.**	04h
75h	RO	**Report no second DMA assignment.**	04h
F0h	R/W	**Parallel Port Configuration register.** (See Table 4-16.)	F2h

Table 4-16. Parallel Port Configuration Register

Bit	Description
Index F0h **Parallel Port Configuration Register (R/W)** **Reset Value: F2h** This register is reset by hardware to F2h.	
7:5	**Reserved.** Must be 11.
4	**Extended Register Access.** 0: Registers at base (address)+403h, base+404h and base+405h are not accessible (reads and writes are ignored). 1: Registers at base (address)+403h, base+404h and base+405h are accessible. This option supports run-time configuration within the Parallel Port address space.
3:2	**Reserved.**
1	**Power Mode Control.** When the logical device is active: 0: Parallel port clock disabled. ECP modes and EPP timeout are not functional when the logical device is active. Registers are maintained. 1: Parallel port clock enabled. All operation modes are functional when the logical device is active. (Default)
0	**TRI-STATE Control.** When enabled and the device is inactive, the logical device output pins are in TRI-STATE. 0: Disable. (Default) 1: Enable.

SuperI/O Module (Continued)

4.5 REAL-TIME CLOCK (RTC)

The RTC provides timekeeping and calendar management capabilities. The RTC uses a 32.768 KHz signal as the basic clock for timekeeping. It also includes 242 bytes of battery-backed RAM for general-purpose use.

The RTC provides the following functions:

- Accurate timekeeping and calendar management
- Alarm at a predetermined time and/or date
- Three programmable interrupt sources
- Valid timekeeping during power-down, by utilizing external battery backup
- 242 bytes of battery-backed RAM
- RAM lock schemes to protect its content
- Internal oscillator circuit (the crystal itself is off-chip), or external clock supply for the 32.768 KHz clock
- A century counter
- PnP support:
 — Relocatable Index and Data registers
 — Module access enable/disable option
 — Host interrupt enable/disable option
- Additional low-power features such as:
 — Automatic switching from battery to V_{SB}
 — Internal power monitoring on the VRT bit
 — Oscillator disabling to save battery during storage
- Software compatible with the DS1287 and MC146818

4.5.1 Bus Interface

The RTC function is initially mapped to the default SuperI/O locations at Indexes 70h to 73h (two Index/Data pairs). These locations may be reassigned, in compliance with Plug and Play requirements.

4.5.2 RTC Clock Generation

The RTC uses a 32.768 KHz clock signal as the basic clock for timekeeping. The 32.768 KHz clock can be supplied by the internal oscillator circuit, or by an external oscillator (see Section 4.5.2.2 "External Oscillator" on page 115).

4.5.2.1 Internal Oscillator

The internal oscillator employs an external crystal connected to the on-chip amplifier. The on-chip amplifier is accessible on the X32I input and X32O output. See Figure 4-5 for the recommended external circuit and Table 4-17 for a listing of the circuit components. The oscillator may be disabled in certain conditions. See Section 4.5.2.8 "Oscillator Activity" on page 118 for more details.

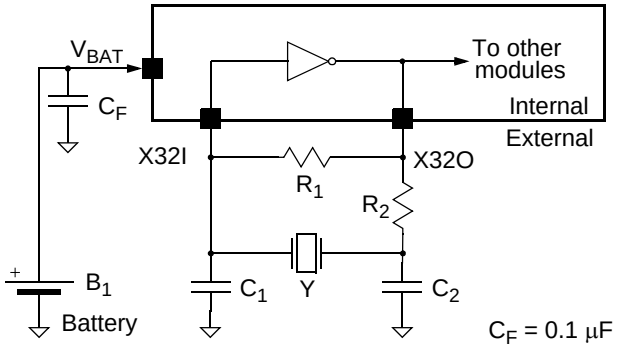


Figure 4-5. Recommended Oscillator External Circuitry

Table 4-17. Crystal Oscillator Circuit Components

Component	Parameters	Values	Tolerance
Crystal	Resonance Frequency	32.768 KHz Parallel mode	User-defined
	Type	N-cut or XY-bar	
	Serial Resistance	40 KΩ	Max
	Quality Factor, Q	35000	Min
	Shunt Capacitance	2 pF	Max
	Load Capacitance, C_L	9-13 pF	
	Temperature Coefficient	User-defined	
Resistor R_1	Resistance	20 MΩ	5%
Resistor R_2	Resistance	120 KΩ	5%
Capacitor C_1	Capacitance	3 to 10 pF	5%
Capacitor C_2	Capacitance	3 to 10 pF	5%

SuperI/O Module (Continued)

External Elements

Choose C_1 and C_2 capacitors (see Figure 4-5 on page 114) to match the crystal's load capacitance. The load capacitance C_L "seen" by crystal Y is comprised of C_1 in series with C_2 and in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout and socket (if any), and can vary from 0 to 10 pF. The rule of thumb in choosing these capacitors is:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PARASITIC}$$

Example:

Crystal C_L = 10 pF, $C_{PARASITIC}$ = 8.2 pF
C_1 = 3.6 pF, C_2 = 3.6 pF

Oscillator Startup

The oscillator starts to generate 32.768 KHz pulses to the RTC after about 100 msec from when V_{BAT} is higher than V_{BATMIN} (2.4V) or V_{SB} is higher than V_{SBMIN} (3.0V). The oscillation amplitude on the X32O pin stabilizes to its final value (approximately 0.4V peak-to-peak around 0.7V DC) in about 1 s.

C_1 can be trimmed to achieve precisely 32.768 KHz. To achieve a high time accuracy, use crystal and capacitors with low tolerance and temperature coefficients.

4.5.2.2 External Oscillator

32.768 KHz can be applied from an external clock source, as shown in Figure 4-6.

Connections

Connect the clock to the X32I ball, leaving the oscillator output, X32O, unconnected.

Signal Parameters

The signal levels should conform to the voltage level requirements for X32I, of square or sine wave of 0.0V to V_{CORE} amplitude. The signal should have a duty cycle of approximately 50%. It should be sourced from a battery-backed source in order to oscillate during power-down. This assures that the RTC delivers updated time/calendar information.

4.5.2.3 Timing Generation

The timing generation function divides the 32.768 KHz clock by 2^{15} to derive a 1 Hz signal, which serves as the input for the seconds counter. This is performed by a divider chain composed of 15 divide-by-two latches, as shown in Figure 4-7.

Bits [6:4] (DV[2:0]) of the CRA Register control the following functions:

- Normal operation of the divider chain (counting).
- Divider chain reset to 0.
- Oscillator activity when only V_{BAT} power is present (backup state).

The divider chain can be activated by setting normal operational mode (bits [6:4] of CRA = 01x or 100). The first update occurs 500 msec after divider chain activation.

Bits [3:0] of CRA select one the of fifteen taps from the divider chain to be used as a periodic interrupt. The periodic flag becomes active after half of the programmed period has elapsed, following divider chain activation.

See Table 4-20 on page 120 for more details.

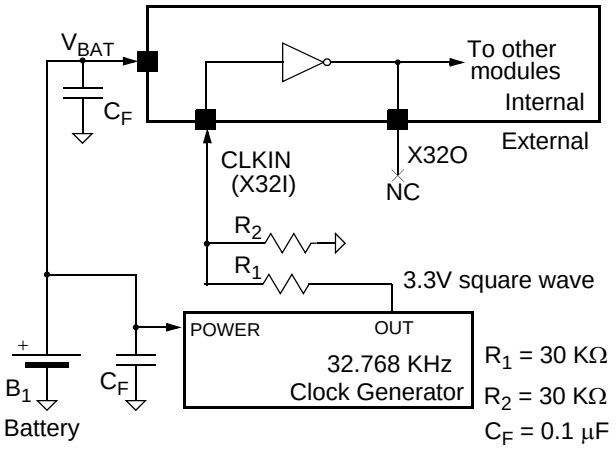


Figure 4-6. External Oscillator Connections

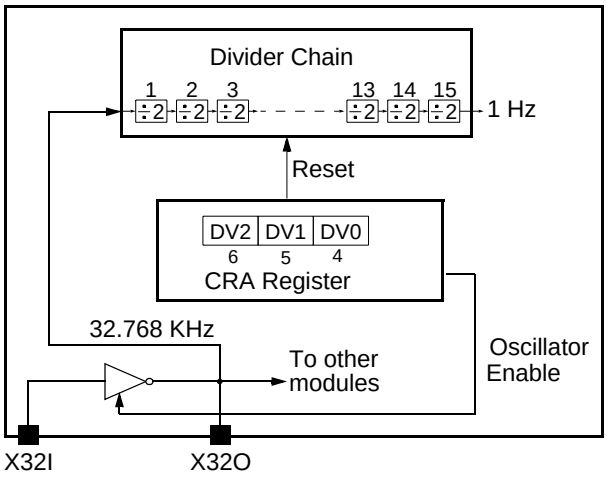


Figure 4-7. Divider Chain Control

SuperI/O Module (Continued)

4.5.2.4 Timekeeping

Data Format

Time is kept in BCD or binary format, as determined by bit 2 (DM) of Control Register B (CRB), and in either 12 or 24-hour format, as determined by bit 1 of this register.

Note: When changing the above formats, re-initialize all the time registers.

Daylight Saving

Daylight saving time exceptions are handled automatically, as described in Table 4-20 on page 120.

Leap Years

Leap year exceptions are handled automatically by the internal calendar function. Every four years, February is extended to 29 days.

Updating

The time and calendar registers are updated once per second regardless of bit 7 (SET) of CRB. Since the time and calendar registers are updated serially, unpredictable results may occur if they are accessed during the update. Therefore, you must ensure that reading or writing to the time storage locations does not coincide with a system update of these locations. There are several methods to avoid this contention.

Method 1

1) Set bit 7 of CRB to 1. This takes a "snapshot" of the internal time registers and loads them into the user copy registers. The user copy registers are seen when accessing the RTC from outside, and are part of the double buffering mechanism. You may keep this bit set for up to 1 second, since the time/calendar chain continue to be updated once per second.
2) Read or write the required registers (since bit 1 is set, you are accessing the user copy registers). If you perform a read operation, the information you read is correct from the time when bit 1 was set. If you perform a write operation, you write only to the user copy registers.
3) Reset bit 1 to 0. During the transition, the user copy registers update the internal registers, using the double buffering mechanism to ensure that the update is performed between two time updates. This mechanism enables new time parameters to be loaded in the RTC.

Method 2

1) Access the RTC registers after detection of an Update Ended interrupt. This implies that an update has just been completed and 999 msec remain until the next update.
2) To detect an Update Ended interrupt, you may either:
 - Poll bit 4 of CRC.
 - Use the following interrupt routine:
 - Set bit 4 of CRB.
 - Wait for an interrupt from interrupt pin.
 - Clear the IRQF flag of CRC before exiting the interrupt routine.

Method 3

Poll bit 7 of CRA. The update occurs 244 μs after this bit goes high. Therefore, if a 0 is read, the time registers remain stable for at least 244 μs.

Method 4

Use a periodic interrupt routine to determine if an update cycle is in progress, as follows:

1) Set the periodic interrupt to the desired period.
2) Set bit 6 of CRB to enable the interrupt from periodic interrupt.
3) Wait for the periodic interrupt appearance. This indicates that the period represented by the following expression remains until another update occurs:
 [(Period of periodic interrupt / 2) + 244 μs]

4.5.2.5 Alarms

The timekeeping function can be set to generate an alarm when the current time reaches a stored alarm time. After each RTC time update (every 1 second), the seconds, minutes, hours, date of month and month counters are compared with their corresponding registers in the alarm settings. If equal, bit 5 of CRC is set. If the Alarm Interrupt Enable bit was previously set (CRB bit 5), interrupt request pin is also active.

Any alarm register may be set to "Unconditional Match" by setting bits [7:6] to 11. This combination, not used by any BCD or binary time codes, results in a periodic alarm. The rate of this periodic alarm is determined by the registers that were set to "Unconditional Match".

For example, if all but the seconds and minutes alarm registers are set to "Unconditional Match", an interrupt is generated every hour at the specified minute and second. If all but the seconds, minutes and hours alarm registers are set to "Unconditional Match", an interrupt is generated every day at the specified hour, minute and second.

SuperI/O Module (Continued)

4.5.2.6 Power Supply

The device is supplied from two supply voltages, as shown in Figure 4-8:

- System standby power supply voltage, V_{SB}
- Backup voltage, from low capacity Lithium battery

A standby voltage, V_{SB}, from the external AC/DC power supply powers the RTC under normal conditions.

Figure 4-9 represents a typical battery configuration. No external diode is required to meet the UL standard, due to the internal switch and internal serial resistor R_{UL}.

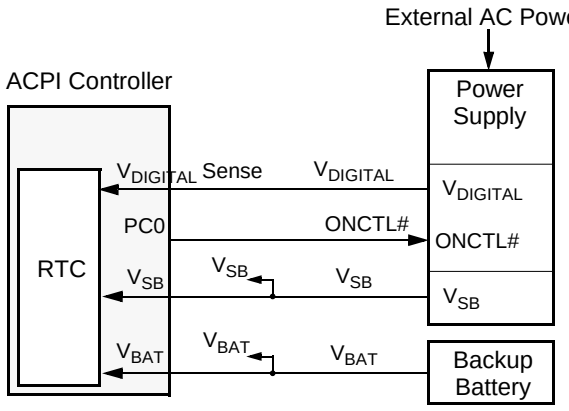
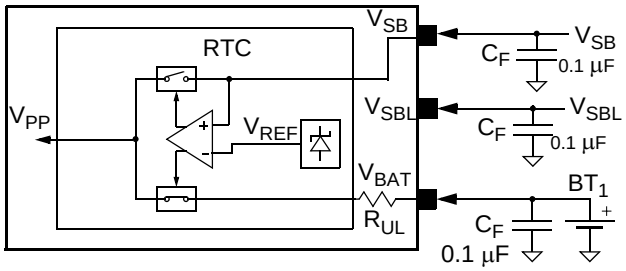


Figure 4-8. Power Supply Connections



Note: Place a 0.1 µF capacitor on each V_{SB}, V_{SBL} power supply pin as close as possible to the pin, and also on V_{BAT}.

Figure 4-9. Typical Battery Configuration

The RTC is supplied from one of two power supplies, V_{SB} or V_{BAT}, according to their levels. An internal voltage comparator delivers the control signals to a pair of switches. Battery backup voltage V_{BAT} maintains the correct time and saves the CMOS memory when the V_{SB} voltage is absent, due to power failure or disconnection of the external AC/DC input power supply or V_{SB} main battery.

To assure that the module uses power from V_{SB} and not from V_{BAT}, the V_{SB} voltage should be maintained above its minimum, as detailed in Section 8.0 "Electrical Specifications" on page 358.

The actual voltage point where the module switches from V_{BAT} to V_{SB} is lower than the minimum workable battery voltage, but high enough to guarantee the correct functionality of the oscillator and the CMOS RAM.

Figure 4-10 shows typical battery current consumption during battery-backed operation, and Figure 4-11 during normal operation.

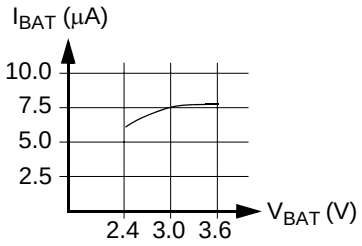
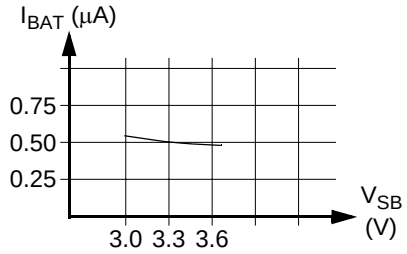


Figure 4-10. Typical Battery Current: Battery Backed Power Mode @ T_C = 25°C



Note: Battery voltage in this test is 3.0V.

Figure 4-11. Typical Battery Current: Normal Operation Mode

SuperI/O Module (Continued)

4.5.2.7 System Power States

The system power state may be No Power, Power On, Power Off or Power Failure. Table 4-18 indicates the power-source combinations for each state. No other power-source combinations are valid.

In addition, the power sources and distribution for the entire system are illustrated in Figure 4-8 on page 117.

Table 4-18. System Power States

$V_{DIGITAL}$	V_{SB}	V_{BAT}	Power State
–	–	–	No Power
–	–	+	Power Failure
–	+	+ or -	Power Off
+	+	+ or -	Power On

No Power

This state exists when no external or battery power is connected to the device. This condition does not occur once a backup battery has been connected, except in the case of a malfunction.

Power On

This is the normal state when the system is active. This state may be initiated by various events in addition to the normal physical switching on of the system. In this state, the system power supply is powered by external AC power and produces $V_{DIGITAL}$ and V_{SB}. The system and the part are powered by $V_{DIGITAL,}$ with the exception of the RTC logical device, which is powered by $V_{SB.}$

Power Off (Suspended)

This is the normal state when the system has been switched off and is not required to be active, but is still connected to a live external AC input power source. This state may be initiated directly or by software. The system is powered down. The RTC logical device remains active, powered by V_{SB}.

Power Failure

This state occurs when the external power source to the system stops supplying power, due to disconnection or power failure on the external AC input power source. The RTC continues to maintain timekeeping and RAM data under battery power (V_{BAT}), unless the oscillator stop bit was set in the RTC. In this case, the oscillator stops functioning if the system goes to battery power, and timekeeping data becomes invalid.

System Bus Lockout

During power on or power off, spurious bus transactions from the host may occur. To protect the RTC internal registers from corruption, all inputs are automatically locked out. The lockout condition is asserted when V_{SB} is lower than V_{SBON}.

Power-Up Detection

When system power is restored after a power failure or power off state (V_{SB} = 0), the lockout condition continues for a delay of 62 msec (minimum) to 125 msec (maximum) after the RTC switches from battery to system power.

The lockout condition is switched off immediately in the following situations:

- If the Divider Chain Control bits, DV[2:0], (CRA bits [6:4]) specify a normal operation mode (01x or 100), all input signals are enabled immediately upon detection of system voltage above V_{SBON}.
- When battery voltage is below V_{BATDCT} and HMR is 1, all input signals are enabled immediately upon detection of system voltage above V_{SBON}. This also initializes registers at offsets 00h through 0Dh.
- If bit 7 (VRT) of CRD is 0, all input signals are enabled immediately upon detection of system voltage above V_{SBON}.

4.5.2.8 Oscillator Activity

The RTC oscillator is active if:

- V_{SB} power supply is higher than V_{SBON}, independent of the battery voltage, V_{BAT}

-or-

- V_{BAT} power supply is higher than V_{BATMIN}, regardless if V_{SB} is present or not.

The RTC oscillator is disabled if:

- During power-down (V_{BAT} only), the battery voltage drops below V_{BATMIN}. When this occurs, the oscillator may be disabled and its functionality cannot be guaranteed.

-or-

- Software wrote 00x to DV[2:0] bits of the CRA Register and V_{SB} is removed. This disables the oscillator and decreases the power consumption from the battery connected to V_{BAT}. When disabling the oscillator, the CMOS RAM is not affected as long as the battery is present at a correct voltage level.

If the RTC oscillator becomes inactive, the following features are dysfunctional/disabled:

- Timekeeping
- Periodic interrupt
- Alarm

SuperI/O Module (Continued)

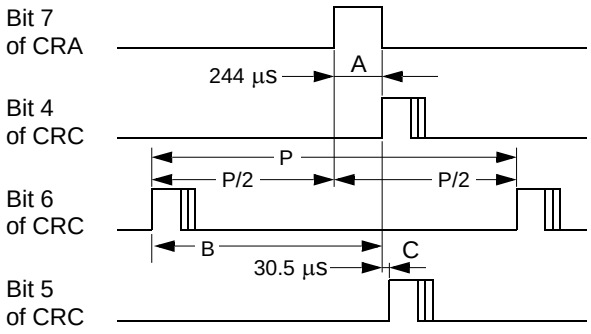
4.5.2.9 Interrupt Handling

The RTC has a single Interrupt Request line which handles the following three interrupt conditions:

- Periodic interrupt
- Alarm interrupt
- Update end interrupt

The interrupts are generated if the respective enable bits in the CRB register are set prior to an interrupt event occurrence. Reading the CRC register clears all interrupt flags. Thus, when multiple interrupts are enabled, the interrupt service routine should first read and store the CRC register, and then deal with all pending interrupts by referring to this stored status.

If an interrupt is not serviced before a second occurrence of the same interrupt condition, the second interrupt event is lost. Figure 4-12 illustrates the interrupt timing in the RTC.



Flags (and IRQ) are reset at the conclusion of CRC read or by reset.

A = Update In Progress bit high before update occurs = 244 µs

B = Periodic interrupt to update = Period (periodic int) / 2 + 244 µs

C = Update to Alarm Interrupt = 30.5 µs

P = Period is programmed by RS[3:0] of CRA

Figure 4-12. Interrupt/Status Timing

4.5.2.10 Battery-Backed RAMs and Registers

The RTC has two battery-backed RAMs and 17 registers, used by the logical units themselves. Battery-backup power enables information retention during system power down.

The RAMs are:

- Standard RAM
- Extended RAM

The memory maps and register content of the RAMs is provided in Section 4.5.4 "RTC General-Purpose RAM Map" on page 124.

The first 14 bytes and 3 programmable bytes of the Standard RAM are overlaid by time, alarm data and control registers. The remaining 111 bytes are general-purpose memory.

Registers with reserved bits should be written using the read-modify-write method.

All register locations within the device are accessed by the RTC Index and Data registers (at base address and base address+1). The Index register points to the register location being accessed, and the Data register contains the data to be transferred to or from the location. An additional 128 bytes of battery-backed RAM (also called Extended RAM) may be accessed via a second pair of Index and Data registers.

Access to the two RAMs may be locked. For details see Table 4-7 on page 108.

SuperI/O Module (Continued)

4.5.3 RTC Registers

The RTC registers can be accessed (see Section 4.4.2.1 "LDN 00h - Real-Time Clock" on page 107) at any time during normal operation mode (i.e.,when V_{SB} is within the recommended operation range). This access is disabled during battery-backed operation. The write operation to these registers is also disabled if bit 7 of the CRD Register is 0.

Note: Before attempting to perform any start-up procedures, read about bit 7 (VRT) of the CRD Register.

This section describes the RTC Timing and Control Registers that control basic RTC functionality.

Table 4-19. RTC Register Map

Index	Type	Name	Reset Type
00h	R/W	**SEC.** Seconds Register	V_{PP} PUR
01h	R/W	**SECA.** Seconds Alarm Register	V_{PP} PUR
02h	R/W	**MIN.** Minutes Register	V_{PP} PUR
03h	R/W	**MINA.** Minutes Alarm Register	V_{PP} PUR
04h	R/W	**HOR.** Hours Register	V_{PP} PUR
05h	R/W	**HORA.** Hours Alarm Register	V_{PP} PUR
06h	R/W	**DOW.** Day Of Week Register	V_{PP} PUR
07h	R/W	**DOM.** Date Of Month Register	V_{PP} PUR
08h	R/W	**MON.** Month Register	V_{PP} PUR
09h	R/W	**YER.** Year Register	V_{PP} PUR
0Ah	R/W	**CRA.** RTC Control Register A	Bit specific
0Bh	R/W	**CRB.** RTC Control Register B	Bit specific
0Ch	RO	**CRC.** RTC Control Register C	Bit specific
0Dh	RO	**CRD.** RTC Control Register D	V_{PP} PUR
Programmable[1]	R/W	**DOMA.** Date of Month Alarm Register	V_{PP} PUR
Programmable[1]	R/W	**MONA.** Month Alarm Register	V_{PP} PUR
Programmable[1]	R/W	**CEN.** Century Register	V_{PP} PUR

1. Overlaid on RAM bytes in range 0Eh-7Fh. See Section 4.4.2.1 "LDN 00h - Real-Time Clock" on page 107.

Table 4-20. RTC Registers

Bit	Description	
Index 00h	**Seconds Register - SEC (R/W)**	**Reset Type: V_{PP} PUR**
7:0	**Seconds Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format.	
Index 01h	**Seconds Alarm Register - SECA (R/W)**	**Reset Type: V_{PP} PUR**
7:0	**Seconds Alarm Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected.	
Index 02h	**Minutes Register - MIN (R/W)**	**Reset Type: V_{PP} PUR**
7:0	**Minutes Data.** Values can be 00 to 59 in BCD format, or 00 to 3B in binary format.	
Index 03h	**Minutes Alarm Register - MINA (R/W)**	**Reset Type: V_{PP} PUR**
7:0	**Minutes Alarm Data.** Values can be 00 to 59 in BCD format, or 00 to 3B in binary format. When bits 7 and 6 are both set to 1, unconditional match is selected. See Section 4.5.2.5 "Alarms" on page 116 for more information about “unconditional” matches.	

SuperI/O Module (Continued)

Table 4-20. RTC Registers (Continued)

Bit	Description
Index 04h	**Hours Register - HOR (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Hours Data.** For 12-hour mode, values can be 01 to 12 (AM) and 81 to 92 (PM) in BCD format, or 01 to 0C (AM) and 81 to 8C (PM) in binary format. For 24-hour mode, values can be 0- to 23 in BCD format or 00 to 17 in binary format.
Index 05h	**Hours Alarm Register - HORA (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Hours Alarm Data.** For 12-hour mode, values may be 01 to 12 (AM) and 81 to 92 (PM) in BCD format or 01 to 0C (AM) and 81 to 8C (PM) in Binary format. For 24-hour mode, values may be 0- to 23 in BCD format or 00 to 17 in Binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected.
Index 06h	**Day of Week Register - DOW (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Day Of Week Data.** Values may be 01 to 07 in BCD format or 01 to 07 in binary format.
Index 07h	**Date of Month Register - DOM (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Date Of Month Data.** Values may be 01 to 31 in BCD format or 01 to 1F in binary format.
Index 08h	**Month Register - MON (R/W)** **Reset Type: V_{PP} PUR**
Width: Byte	
7-0	**Month Data.** Values may be 01 to 12 in BCD format or 01 to 0C in binary format.
Index 09h	**Year Register - YER (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Year Data.** Values may be 00 to 99 in BCD format or 00 to 63 in binary format.
Index 0Ah	**RTC Control Register A - CRA (R/W)** **Reset Type: Bit Specific**
This register controls test selection, among other functions. This register cannot be written before reading bit 7 of CRD.	
7	**Update in Progress. (RO)** This bit is not affected by reset. This bit reads 0 when bit 7 of the CRB Register is 1. 0: Timing registers not updated within 244 μs. 1: Timing registers updated within 244 μs.
6:4	**Divider Chain Control.** These bits control the configuration of the divider chain for timing generation and register bank selection. See Table 4-21 on page 123. They are cleared to 000 as long as bit 7 of CRD is 0.
3:0	**Periodic Interrupt Rate Select.** These bits select one of fifteen output taps from the clock divider chain to control the rate of the periodic interrupt. See Table 4-22 on page 123 and Figure 4-7 on page 115. They are cleared to 000 as long as bit 7 of CRD is 0.
Index 0Bh	**RTC Control Register B - CRB (R/W)** **Reset Type: Bit Specific**
7	**Set Mode.** This bit is reset at V_{PP} power-up reset only. 0: Timing updates occur normally. 1: User copy of time is “frozen”, allowing the time registers to be accessed whether or not an update occurs.
6	**Periodic Interrupt.** Bits [3:0] of the CRA Register determine the rate at which this interrupt is generated. It is cleared to 0 on RTC reset (i.e., hardware or software reset) or when RTC is disable. 0: Disable. 1: Enable.
5	**Alarm Interrupt.** This interrupt is generated immediately after a time update in which the seconds, minutes, hours, date and month time equal their respective alarm counterparts. It is cleared to 0 as long as bit 7 of the CRD Register is reads 0. 0: Disable. 1: Enable.
4	**Update Ended Interrupt.** This interrupt is generated when an update occurs. It is cleared to 0 on RTC reset (i.e., hardware or software reset) or when the RTC is disable. 0: Disable. 1: Enable.
3	**Reserved.** This bit is defined as “Square Wave Enable” by the MC146818 and is not supported by the RTC. This bit is always read as 0.
2	**Data Mode.** This bit is reset at V_{PP} power-up reset only. 0: Enable BCD format. 1: Enable Binary format.

SuperI/O Module (Continued)

Table 4-20. RTC Registers (Continued)

Bit	Description
1	**Hour Mode.** This bit is reset at V_{PP} power-up reset only. 0: Enable 12-hour format. 1: Enable 24-hour format.
0	**Daylight Saving.** This bit is reset at V_{PP} power-up reset only. 0: Disable. 1: Enable: - In the spring, time advances from1:59:59 AM to 3:00:00 AM on the first Sunday in April. - In the fall, time returns from 1:59:59 AM to 1:00:00 AM on the last Sunday in October.
Index 0Ch	**RTC Control Register C - CRC (RO)** — **Reset Type: Bit Specific**
7	**IRQ Flag.** Mirrors the value on the interrupt output signal. When interrupt is active, IRQF is 1. To clear this bit (and deactivate the interrupt pin), read the CRC Register as the flag bits UF, AF and PF are cleared after reading this register. 0: IRQ inactive. 1: Logic equation is true: ((UIE and UF) or (AIE and AF) or (PIE and PF)).
6	**Periodic Interrupt Flag.** Cleared to 0 on RTC reset (i.e., hardware or software reset) or the RTC disabled. In addition, this bit is cleared to 0 when this register is read. 0: No transition occurred on the selected tap since the last read. 1: Transition occurred on the selected tap of the divider chain.
5	**Alarm Interrupt Flag.** Cleared to 0 as long as bit 7 of the CRD Register is reads 0. In addition, this bit is cleared to 0 when this register is read. 0: No alarm detected since the last read. 1: Alarm condition detected.
4	**Update Ended Interrupt Flag.** Cleared to 0 on RTC reset (i.e., hardware or software reset) or the RTC disabled. In addition, this bit is cleared to 0 when this register is read. 0: No update occurred since the last read. 1: Time registers updated.
3:0	**Reserved.**
Index 0Dh	**RTC Control Register D - CRD (RO)** — **Reset Type: V_{PP} PUR**
7	**Valid RAM and Time.** This bit senses the voltage that feeds the RTC (VSB or VBAT) and indicates whether or not it was too low since the last time this bit was read. If it was too low, the RTC contents (time/calendar registers and CMOS RAM) is not valid. 0: The voltage that feeds the RTC was too low. 1: RTC contents (time/calendar registers and CMOS RAM) are valid.
6:0	**Reserved.**
Index Programmable	**Date of Month Alarm Register - DOMA (R/W)** — **Reset Type: V_{PP} PUR**
7:0	**Date of Month Alarm Data**. Values may be 01 to 31 in BCD format or 01 to 1F in Binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected. (Default)
Index Programmable	**Month Alarm Register - MONA (R/W)** — **Reset Type: V_{PP} PUR**
7:0	**Month Alarm Data.** Values may be 01 to 12 in BCD format or 01 to 0C in Binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected. (Default)
Index Programmable	**Century Register - CEN (R/W)** — **Reset Type: V_{PP} PUR**
7:0	**Century Data.** Values may be 00 to 99 in BCD format or 00 to 63 in Binary format.

SuperI/O Module (Continued)

Table 4-21. Divider Chain Control / Test Selection

DV2	DV1	DV0	
CRA6	CRA5	CRA4	Configuration
0	0	X	Oscillator Disabled
0	1	0	Normal Operation
0	1	1	Test
1	0	X	
1	1	X	Divider Chain Reset

Table 4-22. Periodic Interrupt Rate Encoding

Rate Select 3 2 1 0	Periodic Interrupt Rate (msec)	Divider Chain Output
0 0 0 0	No interrupts	
0 0 0 1	3.906250	7
0 0 1 0	7.812500	8
0 0 1 1	0.122070	2
0 1 0 0	0.244141	3
0 1 0 1	0.488281	4
0 1 1 0	0.976562	5
0 1 1 1	1.953125	6
1 0 0 0	3.906250	7
1 0 0 1	7.812500	8
1 0 1 0	15.625000	9
1 0 1 1	31.250000	10
1 1 0 0	62.500000	11
1 1 0 1	125.000000	12
1 1 1 0	250.000000	13
1 1 1 1	500.000000	14

Table 4-23. BCD and Binary Formats

Parameter	BCD Format	Binary Format
Seconds	00 to 59	00 to 3B
Minutes	00 to 59	00 to 3B
Hours	12-hour mode: 01 to 12 (AM) 81 to 92 (PM) 24-hour mode: 00 to 23	12-hour mode: 01 to 0C (AM) 81 to 8C (PM) 24-hour mode: 00 to 17
Day	01 to 07 (Sunday = 01)	01 to 07
Date	01 to 31	01 to 1F
Month	01 to 12 (January = 01)	01 to 0C
Year	00 to 99	00 to 63
Century	00 to 99	00 to 63

4.5.3.1 Usage Hints

1) Read bit 7 of CRD at each system power-up to validate the contents of the RTC registers and the CMOS RAM. When this bit is 0, the contents of these registers and the CMOS RAM are questionable. This bit is reset when the backup battery voltage is too low. The voltage level at which this bit is reset is below the minimum recommended battery voltage, 2.4V. Although the RTC oscillator may function properly and the register contents may be correct at lower than 2.4V, this bit is reset since correct functionality cannot be guaranteed. System BIOS may use a checksum method to revalidate the contents of the CMOS-RAM. The checksum byte should be stored in the same CMOS RAM.
2) Change the backup battery while normal operating power is present, and not in backup mode, to maintain valid time and register information. If a low leakage capacitor is connected to V_{BAT}, the battery may be changed in backup mode.
3) A rechargeable NiCd battery may be used instead of a non-rechargeable Lithium battery. This is a preferred solution for portable systems, where small size components is essential.
4) A supercap capacitor may be used instead of the normal Lithium battery. In a portable system usually the V_{SB} voltage is always present since the power management stops the system before its voltage falls to low. The supercap capacitor in the range of 0.047-0.47 F should supply the power during the battery replacement.

4.5.4 RTC General-Purpose RAM Map

Table 4-24. Standard RAM Map

Index	Description
0Eh - 7Fh	Battery-backed general-purpose 111-byte RAM.

Table 4-25. Extended RAM Map

Index	Description
00h - 7Fh	Battery-backed general-purpose 128-byte RAM.

4.6 SYSTEM WAKEUP CONTROL (SWC)

The SWC wakes up the system by sending a power-up request to the ACPI controller in response to the following maskable system events:

- Modem ring (RI2#)
- Audio Codec event (SDATA_IN2)
- Programmable Consumer Electronics IR (CEIR) address

Each system event that is monitored by the SWC is fed into a dedicated detector that decides when the event is active, according to predetermined (either fixed or programmable) criteria. A set of dedicated registers is used to determine the wakeup criteria, including the CEIR address.

A Wakeup Events Status Register (WKSR) and a Wakeup Events Control Register (WKCR) hold a Status bit and Enable bit, respectively, for each possible wakeup event.

Upon detection of an active event, the corresponding Status bit is set to 1. If the event is enabled (the corresponding Enable bit is set to 1), a power-up request is issued to the ACPI controller. In addition, detection of an active wakeup event may be also routed to an arbitrary IRQ.

Disabling an event prevents it from issuing power-up requests, but does not affect the Status bits. A power-up reset is issued to the ACPI controller when both the Status and Enable bits are set to 1 for at least one event type.

SWC logic is powered by V_{SB}. The SWC control and configuration registers are battery backed, powered by V_{PP}. The setup of the wakeup events, including programmable sequences, is retained throughout power failures (no V_{SB}) as long as the battery is connected. V_{PP} is taken from V_{SB} if V_{SB} > 2.0; otherwise, V_{BAT} is used as the V_{PP} source.

Hardware reset does not affect the SWC registers. They are reset only by a SIO software reset or power-up of V_{PP}.

4.6.1 Event Detection

4.6.1.1 Audio Codec Event

A low-to-high transition on SDATA_IN2 indicates the detection of an Audio Codec event and can be used as a wakeup event.

4.6.1.2 CEIR Address

A CEIR transmission received on IRRX1 in a pre-selected standard (NEC, RCA or RC-5) is matched against a programmable CEIR address. Detection of matching can be used as a wakeup event. The CEIR address detection operates independently of the Serial Port with the IR (which is powered down with the rest of the system).

Whenever an IR signal is detected, the receiver immediately enters the Active state. When this happens, the receiver keeps sampling the IR input signal and generates a bit string where a logic 1 indicates an idle condition and a logic 0 indicates the presence of IR energy. The received bit string is de-serialized and assembled into 8-bit characters.

The expected CEIR protocol of the received signal should be configured through bits [5:4] of the CEIR Wakeup Control register (IRWCR) (see Table 4-30 on page 128).

The CEIR Wakeup Address register (IRWAD) holds the unique address to be compared with the address contained in the incoming CEIR message. If CEIR is enabled (IRWCR[0] = 1) and an address match occurs, then the CEIR Event Status bit of WKSR is set to 1.

The CEIR Address Shift register (ADSR) holds the received address which is compared with the address contained in the IRWAD. The comparison is affected also by the CEIR Wakeup Address Mask register (IRWAM) in which each bit determines whether to ignore the corresponding bit in the IRWAD.

If CEIR routing to interrupt request is enabled, the assigned SWC interrupt request can be used to indicate that a complete address has been received. To get this interrupt when the address is completely received, IRWAM should be written with FFh. Once the interrupt is received, the value of the address can be read from ADSR.

Another parameter that is used to determine whether a CEIR signal is to be considered valid is the bit cell time width. There are four time ranges for the different protocols and carrier frequencies. Four pairs of registers (IRWTRxL and IRWTRxH) define the low and high limits of each time range. Table 4-26 lists the recommended time ranges limits for the different protocols and their applicable ranges. The values are represented in hexadecimal code where the units are of 0.1 ms.

Table 4-26. Time Range Limits for CEIR Protocols

Time Range	RC-5		NEC		RCA	
	Low Limit	High Limit	Low Limit	High Limit	Low Limit	High Limit
0	10h	14h	09h	0Dh	0Ch	12h
1	07h	0Bh	14h	19h	16h	1Ch
2	-	-	50h	64h	B4h	DCh
3	-	-	28h	32h	23h	2Dh

SuperI/O Module (Continued)

4.6.2 SWC Registers

The SWC registers are organized in two banks. The offsets are related to a base address that is determined by the SWC Base Address Register in the logical device configuration. The lower three registers are common to the two banks while the upper registers (03h-0Fh) are divided as follows:

- Bank 0 holds reserved registers.
- Bank 1 holds the CEIR Control Registers.

The active bank is selected through the Configuration Bank Select field (bits [1:0]) in the Wakeup Configuration Register (WKCFG). See Table 4-29 on page 127.

The tables that follow provide register maps and bit definitions for Banks 0 and 1.

Table 4-27. Banks 0 and 1 - Common Control and Status Register Map

Offset	Type	Name	Reset Value
00h	R/W1C	**WKSR.** Wakeup Events Status Register	00h
01h	R/W	**WKCR.** Wakeup Events Control Register	03h
02h	R/W	**WKCFG.** Wakeup Configuration Register	00h

Table 4-28. Bank 1 - CEIR Wakeup Configuration and Control Register Map

Offset	Type	Name	Reset Value
03h	R/W	**IRWCR.** CEIR Wakeup Control Register	00h
04h	---	**RSVD.** Reserved	---
05h	R/W	**IRWAD.** CEIR Wakeup Address Register	00h
06h	R/W	**IRWAM.** CEIR Wakeup Address Mask Register	E0h
07h	RO	**ADSR.** CEIR Address Shift Register	00h
08h	R/W	**IRWTR0L.** CEIR Wakeup, Range 0, Low Limit Register	10h
09h	R/W	**IRWTR0H.** CEIR Wakeup, Range 0, High Limit Register	14h
0Ah	R/W	**IRWTR1L.** CEIR Wakeup, Range 1, Low Limit Register	07h
0Bh	R/W	**IRWTR1H.** CEIR Wakeup, Range 1, High Limit Register	0Bh
0Ch	R/W	**IRWTR2L.** CEIR Wakeup, Range 2, Low Limit Register	50h
0Dh	R/W	**IRWTR2H.** CEIR Wakeup, Range 2, High Limit Register	64h
0Eh	R/W	**IRWTR3L.** CEIR Wakeup, Range 3, Low Limit Register	28h
0Fh	R/W	**IRWTR3H.** CEIR Wakeup, Range 3, High Limit Register	32h

Table 4-29. Banks 0 and 1 - Common Control and Status Registers

Bit	Description
Offset 00h	**Wakeup Events Status Register - WKSR (R/W1C)** **Reset Value: 00h**
This register is set to 00h on power-up of V_{PP} or software reset. It indicates which wakeup event and/or PME occurred. (See Section 5.2.9.4 "Power Management Events" on page 169.)	
7	**Reserved.**
6	**Reserved.** Must be set to 0.
5	**IRRX1 (CEIR) Event Status.** This sticky bit shows the status of the CEIR event detection. 0: Event not detected. (Default) 1: Event detected.
4:2	**Reserved.**
1	**RI2# Event Status.** This sticky bit shows the status of RI2# event detection. 0: Event not detected. (Default) 1: Event detected.
0	**SDATA_IN2 Event Status.** This sticky bit shows the status of Audio Codec event detection. 0: Event not detected. (Default) 1: Event detected.
Offset 01h	**Wakeup Events Control Register - WKCR (R/W)** **Reset Value: 03h**
This register is set to 03h on power-up of V_{PP} or software reset. Detected wakeup events that are enabled issue a power-up request the ACPI controller and/or a PME to the Core Logic module. (See Section 5.2.9.4 "Power Management Events" on page 169.)	
7	**Reserved.**
6	**Reserved.** Must be set to 0.
5	**IRRX1 (CEIR) Event Enable.** 0: Disable. (Default) 1: Enable.
4:2	**Reserved.**
1	**RI2# Event Enable.** 0: Disable. 1: Enable. (Default)
0	**SDATA_IN2 Event Enable.** 0: Disable. 1: Enable. (Default)
Offset 02h	**Wakeup Configuration Register - WKCFG (R/W)** **Reset Value: 00h**
This register is set to 00h on power-up of V_{PP} or software reset. It enables access to CEIR registers.	
7:5	**Reserved.**
4	**Reserved.** Must be set to 0.
3	**Reserved.** Must be set to 0.
2	**Reserved.**
1:0	**Configuration Bank Select Bits.** 00: Only shared registers are accessible. 01: Shared registers and Bank 1 (CEIR) registers are accessible. 10: Bank selected. 11: Reserved.

SuperI/O Module (Continued)

Table 4-30. Bank 1 - CEIR Wakeup Configuration and Control Registers

Bit	Description
Bank 1, Offset 03h — CEIR Wakeup Control Register - IRWCR (R/W) — Reset Value: 00h	This register is set to 00h on power-up of V_{PP} or software reset.
7:6	**Reserved.**
5:4	**CEIR Protocol Select.** 00: RC5. 01: NEC/RCA. 1x: Reserved.
3	**Reserved.**
2	**Invert IRRX Input.** 0: Not inverted. (Default) 1: Inverted.
1	**Reserved.**
0	**CEIR Enable.** 0: Disable. (Default) 1: Enable.
Bank 1, Offset 04h — Reserved	
Bank 1, Offset 05h — CEIR Wakeup Address Register - IRWAD (R/W) — Reset Value: 00h	This register defines the station address to be compared with the address contained in the incoming CEIR message. If CEIR is enabled (bit 0 of the IRWCR register is 1) and an address match occurs, then bit 5 of the WKSR register is set to 1. This register is set to 00h on power-up of V_{PP} or software reset.
7:0	**CEIR Wakeup Address**
Bank 1, Offset 06h — CEIR Wakeup Mask Register - IRWAM (R/W) — Reset Value: E0h	Each bit in this register determines whether the corresponding bit in the IRWAD register takes part in the address comparison. Bits 5, 6, and 7 must be set to 1 if the RC-5 protocol is selected. This register is set to E0h on power-up of V_{PP} or software reset.
7:0	**CEIR Wakeup Address Mask.** • If the corresponding bit is 0, the address bit is not masked (enabled for compare). • If the corresponding bit is 1, the address bit is masked (ignored during compare).
Bank 1, Offset 07h — CEIR Address Shift Register - ADSR (RO) — Reset Value: 00h	This register holds the received address to be compared with the address contained in the IRWAD register. This register is set to 00h on power-up of V_{PP} or software reset.
7:0	**CEIR Address.**
CEIR Wakeup Range 0 Registers	These two registers (IRWTR0L and IRWTR0H) define the low and high limits of time range 0 (see Table 4-26 on page 125). The values are represented in units of 0.1 ms. • RC-5 protocol: The bit cell width must fall within this range for the cell to be considered valid. The nominal cell width is 1.778 msec for a 36 KHz carrier. IRWTR0L and IRWTR0H should be set to 10h and 14h, respectively. (Default) • NEC protocol: The time distance between two consecutive CEIR pulses that encodes a bit value of 0 must fall within this range. The nominal distance for a 0 is 1.125 msec for a 38 KHz carrier. IRWTR0L and IRWTR0H should be set to 09h and 0Dh, respectively.
Bank 1, Offset 08h — IRWTR0L Register (R/W) — Reset Value: 10h	This register is set to 10h on power-up of V_{PP} or software reset.
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 0, Low Limit.**
Bank 1, Offset 09h — IRWTR0H Register (R/W) — Reset Value: 14h	This register is set to 14h on power-up of V_{PP} or software reset.
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 0, High Limit.**

SuperI/O Module (Continued)

Table 4-30. Bank 1 - CEIR Wakeup Configuration and Control Registers (Continued)

Bit	Description
	CEIR Wakeup Range 1 Registers These two registers (IRWTR1L and IRWTR1H) define the low and high limits of time range 1 (see Table 4-26 on page 125). The values are represented in units of 0.1 ms. • RC-5 protocol: The pulse width defining a half-bit cell must fall within this range in order for the cell to be considered valid. The nominal pulse width is 0.889 for a 38 KHz carrier. IRWTR1L and IRWTR1H should be set to 07h and 0Bh, respectively. (Default) • NEC protocol: The time between two consecutive CEIR pulses that encodes a bit value of 1 must fall within this range. The nominal time for a 1 is 2.25 msec for a 36 KHz carrier. IRWTR1L and IRWTR1H should be set to 14h and 19h, respectively.
Bank 1, Offset 0Ah	**IRWTR1L Register (R/W)** — **Reset Value: 07h** This register is set to 07h on power-up of V_{PP} or software reset.
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 1, Low Limit.**
Bank 1, Offset 0Bh	**IRWTR1H Register (R/W)** — **Reset Value: 0Bh** This register is set to 0Bh on power-up of V_{PP} or software reset.
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 1, High Limit.**
	CEIR Wakeup Range 2 Registers These two registers (IRWTR2L and IRWTR2H) define the low and high limits of time range 2 (see Table 4-26 on page 125). The values are represented in units of 0.1 ms. • RC-5 protocol: These registers are not used when the RC-5 protocol is selected. • NEC protocol: The header pulse width must fall within this range in order for the header to be considered valid. The nominal value is 9 msec for a 38 KHz carrier. IRWTR2L and IRWTR2H should be set to 50h and 64h, respectively. (Default)
Bank 1, Offset 0Ch	**IRWTR2L Register (R/W)** — **Reset Value: 50h** This register is set to 50h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 2, Low Limit.**
Bank 1, Offset 0Dh	**IRWTR2H Register (R/W)** — **Reset Value: 64h** This register is set to 64h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 2, High Limit.**
	CEIR Wakeup Range 3 Registers These two registers (IRWTR3L and IRWTR3H) define the low and high limits of time range 3 (see Table 4-26 on page 125). The values are represented in units of 0.1 ms. • RC-5 protocol: These registers are not used when the RC-5 protocol is selected. • NEC protocol: The post header gap width must fall within this range in order for the gap to be considered valid. The nominal value is 4.5 msec for a 36 KHz carrier. IRWTR3L and IRWTR3H should be set to 28h and 32h, respectively. (Default)
Bank 1, Offset 0Eh	**IRWTR3L Register (R/W)** — **Reset Value: 28h** This register is set to 28h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 3, Low Limit.**
Bank 1, Offset 0Fh	**IRWTR3H Register (R/W)** — **Reset Value: 32h** This register is set to 32h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 3, High Limit.**

4.7 ACCESS.BUS INTERFACE

The SC2200 has two ACCESS.bus (ACB) controllers. ACB is a two-wire synchronous serial interface compatible with the ACCESS.bus physical layer, Intel's SMBus, and Philips' I^2C. The ACB can be configured as a bus master or slave, and can maintain bidirectional communication with both multiple master and slave devices. As a slave device, the ACB may issue a request to become the bus master.

The ACB allows easy interfacing to a wide range of low-cost memories and I/O devices, including: EEPROMs, SRAMs, timers, ADC, DAC, clock chips and peripheral drivers.

The ACCESS.bus protocol uses a two-wire interface for bidirectional communication between the ICs connected to the bus. The two interface lines are the Serial Data Line (AB1D and AB2D) and the Serial Clock Line (AB1C and AB2C). (Here after referred to as ABD and ABC unless otherwise specified.) These lines should be connected to a positive supply via an internal or external pull-up resistor, and remain high even when the bus is idle.

Each IC has a unique address and can operate as a transmitter or a receiver (though some peripherals are only receivers).

During data transactions, the master device initiates the transaction, generates the clock signal and terminates the transaction. For example, when the ACB initiates a data transaction with an attached ACCESS.bus compliant peripheral, the ACB becomes the master. When the peripheral responds and transmits data to the ACB, their master/slave (data transaction initiator and clock generator) relationship is unchanged, even though their transmitter/receiver functions are reversed.

This section describes the general ACB functional block. A device may include a different implementation. For device specific implementation, see Section 4.4.2.5 "LDN 05h and 06h - ACCESS.bus Ports 1 and 2" on page 112.

4.7.1 Data Transactions

One data bit is transferred during each clock pulse. Data is sampled during the high state of the serial clock (ABC). Consequently, throughout the clock's high period, the data should remain stable (see Figure 4-13). Any changes on the ABD line during the high state of the ABC and in the middle of a transaction aborts the current transaction. New data should be sent during the low ABC state. This protocol permits a single data line to transfer both command/control information and data, using the synchronous serial clock.

Each data transaction is composed of a Start Condition, a number of byte transfers (set by the software) and a Stop Condition to terminate the transaction. Each byte is transferred with the most significant bit first, and after each byte (8 bits), an Acknowledge signal must follow. The following sections provide further details of this process.

During each clock cycle, the slave can stall the master while it handles the previous data or prepares new data. This can be done for each bit transferred, or on a byte boundary, by the slave holding ABC low to extend the clock-low period. Typically, slaves extend the first clock cycle of a transfer if a byte read has not yet been stored, or if the next byte to be transmitted is not yet ready. Some microcontrollers, with limited hardware support for ACCESS.bus, extend the access after each bit, thus allowing the software to handle this bit.

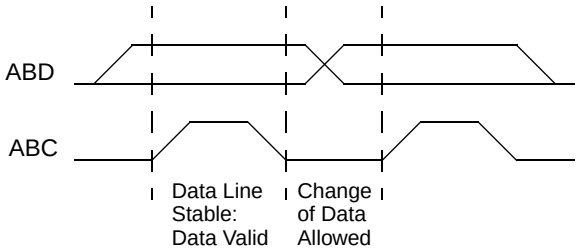


Figure 4-13. Bit Transfer

4.7.2 Start and Stop Conditions

The ACCESS.bus master generates Start and Stop Conditions (control codes). After a Start Condition is generated, the bus is considered busy and retains this status for a certain time after a Stop Condition is generated. A high-to-low transition of the data line (ABD) while the clock (ABC) is high indicates a Start Condition. A low-to-high transition of the ABD line while the ABC is high indicates a Stop Condition (Figure 4-14).

In addition to the first Start Condition, a repeated Start Condition can be generated in the middle of a transaction. This allows another device to be accessed, or a change in the direction of data transfer.

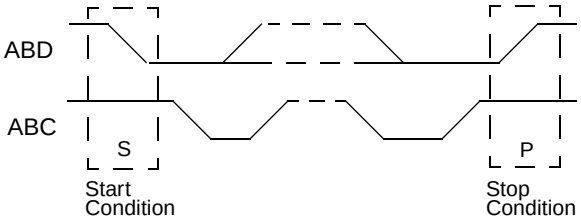


Figure 4-14. Start and Stop Conditions

4.7.3 Acknowledge (ACK) Cycle

The ACK cycle consists of two signals: the ACK clock pulse sent by the master with each byte transferred, and the ACK signal sent by the receiving device (see Figure 4-15).

The master generates the ACK clock pulse on the ninth clock pulse of the byte transfer. The transmitter releases the ABD line (permits it to go high) to allow the receiver to send the ACK signal. The receiver must pull down the ABD line during the ACK clock pulse, signalling that it has correctly received the last data byte and is ready to receive the next byte. Figure 4-16 illustrates the ACK cycle.

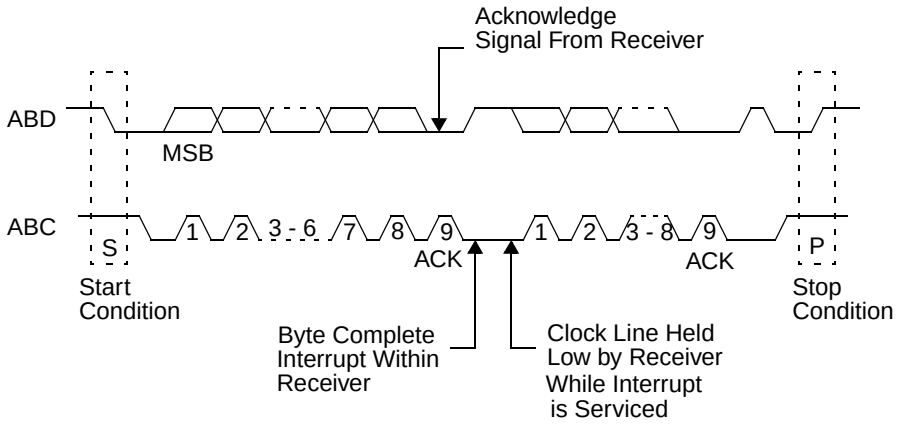


Figure 4-15. ACCESS.bus Data Transaction

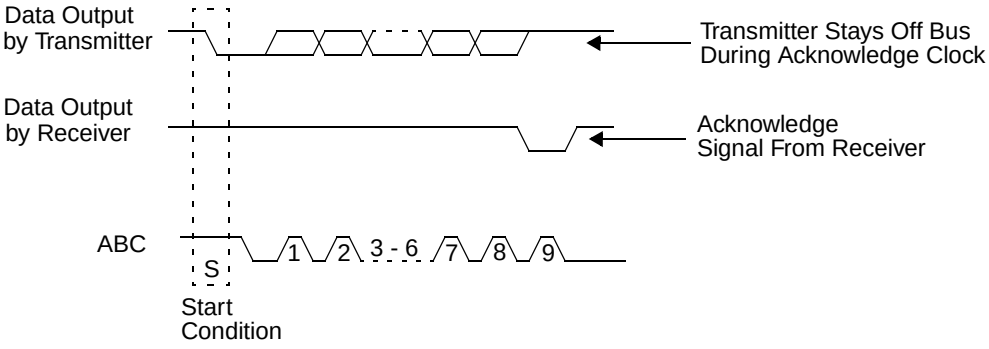


Figure 4-16. ACCESS.bus Acknowledge Cycle

SuperI/O Module (Continued)

4.7.4 Acknowledge After Every Byte Rule

According to this rule, the master generates an acknowledge clock pulse after each byte transfer, and the receiver sends an acknowledge signal after every byte received. There are two exceptions to this rule:

- When the master is the receiver, it must indicate to the transmitter the end of data by not acknowledging (negative acknowledge) the last byte clocked out of the slave. This negative acknowledge still includes the acknowledge clock pulse (generated by the master), but the ABD line is not pulled down.
- When the receiver is full, otherwise occupied, or a problem has occurred, it sends a negative acknowledge to indicate that it cannot accept additional data bytes.

4.7.5 Addressing Transfer Formats

Each device on the bus has a unique address. Before any data is transmitted, the master transmits the address of the slave being addressed. The slave device should send an acknowledge signal on the ABD line, once it recognizes its address.

The address consists of the first 7 bits after a Start Condition. The direction of the data transfer (R/W#) depends on the bit sent after the address, the eighth bit. A low-to-high transition during a ABC high period indicates the Stop Condition, and ends the transaction of ABD (see Figure 4-17).

When the address is sent, each device in the system compares this address with its own. If there is a match, the device considers itself addressed and sends an acknowledge signal. Depending on the state of the R/W# bit (1 = Read, 0 = Write), the device acts either as a transmitter or a receiver.

The I^2C bus protocol allows a general call address to be sent to all slaves connected to the bus. The first byte sent specifies the general call address (00h) and the second byte specifies the meaning of the general call (for example, write slave address by software only). Those slaves that require data acknowledge the call, and become slave receivers; other slaves ignore the call.

4.7.6 Arbitration on the Bus

Multiple master devices on the bus require arbitration between their conflicting bus access demands. Control of the bus is initially determined according to address bits and clock cycle. If the masters are trying to address the same slave, data comparisons determine the outcome of this arbitration. In master mode, the device immediately aborts a transaction if the value sampled on the ABD line differs from the value driven by the device. (An exception to this rule is ABD while receiving data. The lines may be driven low by the slave without causing an abort.)

The ABC signal is monitored for clock synchronization and to allow the slave to stall the bus. The actual clock period is set by the master with the longest clock period, or by the slave stall period. The clock high period is determined by the master with the shortest clock high period.

When an abort occurs during the address transmission, a master that identifies the conflict should give up the bus, switch to slave mode and continue to sample ABD to check if it is being addressed by the winning master on the bus.

4.7.7 Master Mode

Requesting Bus Mastership

An ACCESS.bus transaction starts with a master device requesting bus mastership. It asserts a Start Condition, followed by the address of the device it wants to access. If this transaction is successfully completed, the software may assume that the device has become the bus master.

For the device to become the bus master, the software should perform the following steps:

1) Configure ACBCTL1[2] to the desired operation mode. (Polling or Interrupt) and set the ACBCTL1[0]. This causes the ACB to issue a Start Condition on the ACCESS.bus when the ACCESS.bus becomes free (ACBCST[1] is cleared, or other conditions that can delay start). It then stalls the bus by holding ABC low.
2) If a bus conflict is detected (i.e., another device pulls down the ABC signal), the ACBST[5] is set.
3) If there is no bus conflict, ACBST[1] and ACBST[6] are set.
4) If the ACBCTL1[2] is set and either ACBST[5] or ACBST[6] is set, an interrupt is issued.

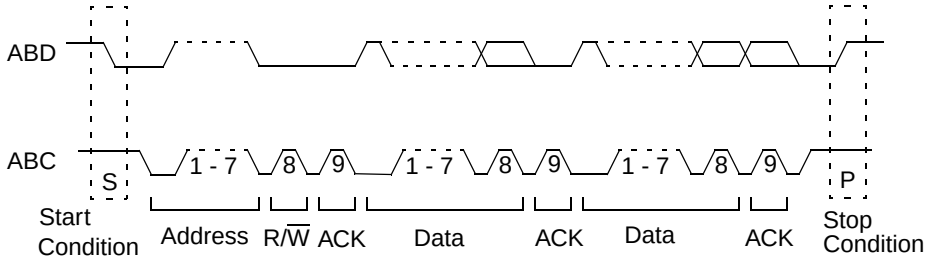


Figure 4-17. A Complete ACCESS.bus Data Transaction

SuperI/O Module (Continued)

Sending the Address Byte

When the device is the active master of the ACCESS.bus (ACBST[1] is set), it can send the address on the bus.

The address sent should not be the device's own address, as defined by ACBADDR[6:0] if ACBADDR[7] is set, nor should it be the global call address if ACBST[3] is set.

To send the address byte, use the following sequence:

1) For a receive transaction where the software wants only one byte of data, it should set ACBCTL1[4]. If only an address needs to be sent or if the device requires stall for some other reason, set ACBCTL1[7].
2) Write the address byte (7-bit target device address) and the direction bit to the ACBSDA register. This causes the ACB to generate a transaction. At the end of this transaction, the acknowledge bit received is copied to ACBST[4]. During the transaction, the ABD and ABC lines are continuously checked for conflict with other devices. If a conflict is detected, the transaction is aborted, ACBST[5] is set and ACBST[1] is cleared.
3) If ACBCTL1[7] is set and the transaction was successfully completed (i.e., both ACBST[5] and ACBST[4] are cleared), ACBST[3] is set. In this case, the ACB stalls any further ACCESS.bus operations (i.e., holds ABC low). If ACBCTL1[2] is set, it also sends an interrupt request to the host.
4) If the requested direction is transmit and the start transaction was completed successfully (i.e., neither ACBST[5] nor ACBST[4] is set, and no other master has accessed the device), ACBST[6] is set to indicate that the ACB awaits attention.
5) If the requested direction is receive, the start transaction was completed successfully and ACBCTL1[7] is cleared, the ACB starts receiving the first byte automatically.
6) Check that both ACBST[5] and ACBST[4] are cleared. If ACBCTL1[2] is set, an interrupt is generated when ACBST[5] or ACBST[4] is set.

Master Transmit

After becoming the bus master, the device can start transmitting data on the ACCESS.bus.

To transmit a byte in an interrupt or polling controlled operation, the software should:

1) Check that both ACBST[5] and ACBST[4] are cleared, and that ACBST[6] is set. If ACBCTL1[7] is set, also check that ACBST[3] is cleared (and clear it if required).
2) Write the data byte to be transmitted to the ACBSDA.

When either ACBST[5] or ACBST[4] is set, an interrupt is generated. When the slave responds with a negative acknowledge, ACBST[4] Register is set and ACBST[6] remains cleared. In this case, if ACBCTL1[2] Register is set, an interrupt is issued.

Master Receive

After becoming the bus master, the device can start receiving data on the ACCESS.bus.

To receive a byte in an interrupt or polling operation, the software should:

1) Check that ACBST[6] is set and that ACBST[5] is cleared. If ACBCTL1[7] is set, also check that the ACBST[3] is cleared (and clear it if required).
2) Set ACBCTL1[4] to 1, if the next byte is the last byte that should be read. This causes a negative acknowledge to be sent.
3) Read the data byte from the ACBSDA.

Before receiving the last byte of data, set ACBCTL1[4].

4.7.7.1 Master Stop

To end a transaction, set the ACBCTL1[1] before clearing the current stall flag (i.e., ACBST[6], ACBST[4], or ACBST[3]). This causes the ACB to send a Stop Condition immediately, and to clear ACBCTL1[1]. A Stop Condition may be issued only when the device is the active bus master (i.e., ACBST[1] is set).

Master Bus Stall

The ACB can stall the ACCESS.bus between transfers while waiting for the host response. The ACCESS.bus is stalled by holding the AB1C signal low after the acknowledge cycle. Note that this is interpreted as the beginning of the following bus operation. The user must make sure that the next operation is prepared before the flag that causes the bus stall is cleared.

The flags that can cause a bus stall in master mode are:

- Negative acknowledge after sending a byte (ACBST[4] = 1).
- ACBST[6] bit is set.
- ACBCTL1[7] = 1, after a successful start (ACBST[3] = 1).

Repeated Start

A repeated start is performed when the device is already the bus master (ACBST[1] is set). In this case, the ACCESS.bus is stalled and the ACB awaits host handling due to: negative acknowledge (ACBST[4] = 1), empty buffer (ACBST[6] = 1) and/or a stall after start (ACBST[3] 1).

For a repeated start:

1) Set \ACBCTL1[0] to 1.
2) In master receive mode, read the last data item from ACBSDA.
3) Follow the address send sequence, as described previously in "Sending the Address Byte". If the ACB was awaiting handling due to ACBST[3] = 1, clear it only after writing the requested address and direction to ACBSDA.

SuperI/O Module (Continued)

Master Error Detection

The ACB detects illegal Start or Stop Conditions (i.e., a Start or Stop Condition within the data transfer, or the acknowledge cycle) and a conflict on the data lines of the ACCESS.bus. If an illegal condition is detected, ACBST[5] is set, and master mode is exited (ACBST[1] is cleared).

Bus Idle Error Recovery

When a request to become the active bus master or a restart operation fails, ACBST[5] is set to indicate the error. In some cases, both the device and the other device may identify the failure and leave the bus idle. In this case, the start sequence may be incomplete and the ACCESS.bus may remain deadlocked.

To recover from deadlock, use the following sequence:

1) Clear ACBST[5] and ACBCST[1].
2) Wait for a timeout period to check that there is no other active master on the bus (i.e., ACBCST[1] remains cleared).
3) Disable, and re-enable the ACB to put it in the non-addressed slave mode. This completely resets the functional block.

At this point, some of the slaves may not identify the bus error. To recover, the ACB becomes the bus master: it asserts a Start Condition, sends an address byte, then asserts a Stop Condition which synchronizes all the slaves.

4.7.8 Slave Mode

A slave device waits in idle mode for a master to initiate a bus transaction. Whenever the ACB is enabled and it is not acting as a master (i.e., ACBST[1] is cleared), it acts as a slave device.

Once a Start Condition on the bus is detected, the device checks whether the address sent by the current master matches either:

- The ACBADDR[6:0] value if ACBADDR[7] = 1.

or

- The general call address if ACBCTL1[5] 1.

This match is checked even when ACBST[1] is set. If a bus conflict (on ABD or ABC) is detected, ACBST[5] is set, ACBST[1] is cleared and the device continues to search the received message for a match.

If an address match or a global match is detected:

1) The device asserts its ABD pin during the acknowledge cycle.
2) ACBCST[2] and ACBST[2] are set. If ACBST[0] = 1 (i.e., slave transmit mode) ACBST[6] is set to indicate that the buffer is empty.
3) If ACBCTL1[2] is set, an interrupt is generated if both ACBCTL1[2] and ACBCTL16 are set.
4) The software then reads ACBST[0] to identify the direction requested by the master device. It clears ACBST[2] so future byte transfers are identified as data bytes.

Slave Receive and Transmit

Slave receive and transmit are performed after a match is detected and the data transfer direction is identified. After a byte transfer, the ACB extends the acknowledge clock until the software reads or writes ACBSDA. The receive and transmit sequences are identical to those used in the master routine.

Slave Bus Stall

When operating as a slave, the device stalls the ACCESS.bus by extending the first clock cycle of a transaction in the following cases:

- ACBST[6] is set.
- ACBST[2] and ACBCTL1[6] are set.

Slave Error Detection

The ACB detects illegal Start and Stop Conditions on the ACCESS.bus (i.e., a Start or Stop Condition within the data transfer or the acknowledge cycle). When this occurs, ACBST[5] is set and ACBCST[3:2] are cleared, setting the ACB as an unaddressed slave.

4.7.9 Configuration

ABD and ABC Signals

The ABD and ABC are open-drain signals. The device permits the user to define whether to enable or disable the internal pull-up of each of these signals.

ACB Clock Frequency

The ACB permits the user to set the clock frequency for the ACCESS.bus clock. The clock is set by the ACBCTL2[7:1], which determines the ABC clock period used by the device. This clock low period may be extended by stall periods initiated by the ACB or by another ACCESS.bus device. In case of a conflict with another bus master, a shorter clock high period may be forced by the other bus master until the conflict is resolved.

SuperI/O Module (Continued)

4.7.10 ACB Registers

Each functional block is associated with a Logical Device Number (LDN) (see Section 4.3.2 "Banked Logical Device Registers" on page 101). ACCESS.Bus Port 1 is assigned as LDN 05h and ACCESS.bus Port 2 as LDN 06h. In addition to the registers listed here, there are additional configuration registers listed in Section 4.4.2.5 "LDN 05h and 06h - ACCESS.bus Ports 1 and 2" on page 112.

Table 4-31. ACB Register Map

Offset	Type	Name	Reset Value
00h	R/W	**ACBSDA.** ACB Serial Data	xxh
01h	R/W	**ACBST.** ACB Status	00h
02h	R/W	**ACBCST.** ACB Control Status	00h
03h	R/W	**ACBCTL1.** ACB Control 1	00h
04h	R/W	**ACBADDR.** ACB Own Address	xxh
05h	R/W	**ACBCTL2.** ACB Control 2	00h

Table 4-32. ACB Registers

Bit	Description
Offset 00h	**ACB Serial Data Register - ACBSDA (R/W)** **Reset Value: xxh**
7:0	**ACB Serial Data.** This shift register is used to transmit and receive data. The most significant bit is transmitted (received) first, and the least significant bit is transmitted last. Reading or writing to ACBSDA is allowed only when ACBST[6] is set, or for repeated starts after setting the ACBCTL1[0]. An attempt to access the register in other cases may produce unpredictable results.
Offset 01h	**ACB Status Register - ACBST (R/W)** **Reset Value: 00h** This is a read register with a special clear. Some of its bits may be cleared by software, as described below. This register maintains the current ACB status. On reset, and when the ACB is disabled, ACBST is cleared (00h).
7	**SLVSTP (Slave Stop). (R/W1C)** Writing 0 to SLVSTP is ignored. 0: Writing 1 or ACB disabled. 1: Stop Condition detected after a slave transfer in which ACBCST[2] or ACBCST[3] was set.
6	**SDAST (SDA Status). (RO)** 0: Reading from ACBSDA during a receive, or when writing to it during a transmit. When ACBCTL1[0] is set, reading ACBSDA does not clear SDAST. This enables ACB to send a repeated start in master receive mode. 1: SDA Data Register awaiting data (transmit - master or slave) or holds data that should be read (receive - master or slave).
5	**BER (Bus Error). (R/W1C)** Writing 0 to this bit is ignored. 0: Writing 1 or ACB disabled. 1: Start or Stop Condition detected during data transfer (i.e., Start or Stop Condition during the transfer of bits [8:2] and acknowledge cycle), or when an arbitration problem detected.
4	**NEGACK (Negative Acknowledge). (R/W1C)** Writing 0 to this bit is ignored. 0: Writing 1 or ACB disabled. 1: Transmission not acknowledged on the ninth clock (In this case, SDAST (bit 6) is not set).
3	**STASTR (Stall After Start). (R/W1C)** Writing 0 to this bit is ignored. 0: Writing 1 or ACB disabled. 1: Address sent successfully (i.e., a Start Condition sent without a bus error, or Negative Acknowledge), if ACBCTL1[7] is set. This bit is ignored in slave mode. When STASTR is set, it stalls the ACCESS.bus by pulling down the ABC line, and suspends any further action on the bus (e.g., receive of first byte in master receive mode). In addition, if ACBCTL1[1] is set, it also causes the ACB to send an interrupt.
2	**NMATCH (New Match). (R/W1C)** Writing 0 to this bit is ignored. If ACBCTL1[2] is set, an interrupt is sent when this bit is set. 0: Software writes 1 to this bit. 1: Address byte follows a Start Condition or a repeated start, causing a match or a global-call match.

SuperI/O Module (Continued)

Table 4-32. ACB Registers (Continued)

Bit	Description
1	**MASTER. (RO)** 0: Arbitration loss (BER, bit 5, is set) or recognition of a Stop Condition. 1: Bus master request succeeded and master mode active.
0	**XMIT (Transmit). (RO)** Direction bit. 0: Master/slave transmit mode not active. 1: Master/slave transmit mode active.
Offset 02h	**ACB Control Status Register - ACBCST (R/W)** **Reset Value: 00h** This register configures and controls the ACB functional block. It maintains the current ACB status and controls several ACB functions. On reset and when the ACB is disabled, the non-reserved bits of ACBCST are cleared.
7:6	**Reserved.**
5	**TGABC (Toggle ABC Line). (R/W)** Enables toggling the ABC line during error recovery. 0: Clock toggle completed. 1: When the ABD line is low, writing 1 to this bit toggles the ABC line for one cycle. Writing 1 to TGABC while ABD is high is ignored.
4	**TSDA (Test ABD Line). (RO)** Reads the current value of the ABD line. It can be used while recovering from an error condition in which the ABD line is constantly pulled low by an out-of-sync slave. Data written to this bit is ignored.
3	**GCMTCH (Global Call Match). (RO)** 0: Start Condition or repeated Start and a Stop Condition (including illegal Start or Stop Condition). 1: In slave mode, ACBCTL1.GCMEN is set and the address byte (the first byte transferred after a Start Condition) is 00h.
2	**MATCH (Address Match). (RO)** 0: Start Condition or repeated Start and a Stop Condition (including illegal Start or Stop Condition). 1: ACBADDR[7] is set and the first 7 bits of the address byte (the first byte transferred after a Start Condition) match the 7-bit address in ACBADDR.
1	**BB (Bus Busy). (R/W1C)** 0: Writing 1, ACB disabled, or Stop Condition detected. 1: Bus active (a low level on either ABD or ABC), or Start Condition.
0	**BUSY. (RO)** This bit should always be written 0. This bit indicates the period between detecting a Start Condition and completing receipt of the address byte. After this, the ACB is either free or enters slave mode. 0: Completion of any state below or ACB disabled. 1: ACB is in one of the following states: -Generating a Start Condition. -Master mode (ACBST[1] is set). -Slave mode (ACBCST[2] or ACBCST[3] set).
Offset 03h	**ACB Control Register 1 - ACBCTL1 (R/W)** **Reset Value: 00h**
7	**STASTRE (Stall After Start Enable).** 0: When cleared, ACBST[3] can not be set. However, if ACBST[3] is set, clearing STASTRE does not clear ACBST[3]. 1: Stall after start mechanism enabled, and ACB stalls the bus after the address byte.
6	**NMINTE (New Match Interrupt Enable).** 0: No interrupt issued on a new match. 1: Interrupt issued on a new match only if ACBCTL1[2] set.
5	**GCMEN (Global Call Match Enable).** 0: Global call match disabled. 1: Global call match enabled.
4	**ACK (Acknowledge).** This bit is ignored in transmit mode. When the device acts as a receiver (slave or master), this bit holds the stop transmitting instruction that is transmitted during the next acknowledge cycle. 0: Cleared after acknowledge cycle. 1: Negative acknowledge issued on next received byte.
3	**Reserved.**

SuperI/O Module (Continued)

Table 4-32. ACB Registers (Continued)

Bit	Description
2	**INTEN (Interrupt Enable).** 0: ACB interrupt disabled. 1: ACB interrupt enabled. An interrupt is generated in response to one of the following events: -Detection of an address match (ACBST[2] = 1) and ACBCTL1[6] = 1. -Receipt of Bus Error (ACBST[5] = 1). -Receipt of Negative Acknowledge after sending a byte (ACBST[4] = 1). -Acknowledge of each transaction (same as the hardware set of the ACBST[6]). -In master mode if ACBCTL1[7] = 1, after a successful start (ACBST[3] = 1). -Detection of a Stop Condition while in slave mode (ACBST[7] = 1).
1	**STOP (Stop).** 0: Automatically cleared after Stop issued. 1: Setting this bit in master mode generates a Stop Condition to complete or abort current message transfer.
0	**START (Start).** Set this bit only when in master mode or when requesting master mode. 0: Cleared after Start Condition sent or Bus Error (ACBST[5] = 1) detected. 1: Single or repeated Start Condition generated on the ACCESS.bus. If the device is not the active master of the bus (ACBST[1] = 0), setting START generates a Start Condition when the ACCESS.bus becomes free (ACBCST[1] = 0). An address transmission sequence should then be performed. If the device is the active master of the bus (ACBST[1] = 1), setting START and then writing to ACBSDA generates a Start Condition. If a transmission is already in progress, a repeated Start Condition is generated. This condition can be used to switch the direction of the data flow between the master and the slave, or to choose another slave device without separating them with a Stop Condition.
Offset 04h	**ACB Own Address Register - ACBADDR (R/W)** **Reset Value: xxh**
7	**SAEN (Slave Address Enable).** 0: ACB does not check for an address match with ACBADDR[6:0]. 1: ACBADDR[6:0] holds a valid address and enables the match of ADDR to an incoming address byte.
6:0	**ADDR (Address).** These bits hold the 7-bit device address of the SC2200. When in slave mode, the first 7 bits received after a Start Condition are compared with this field (first bit received is compared with bit 6, and the last bit with bit 0). If the address field matches the received data and ACBADDR[7] is 1, a match is declared.
Offset 05h	**ACB Control Register 2 - ACBCTL2 (R/W)** **Reset Value: 00h** This register enables/disables the functional block and determines the ACB clock rate.
7:1	**ABCFRQ (ABC Frequency).** This field defines the ABC period (low and high time) when the device serves as a bus master. The clock low and high times are defined as follows: tABCl = tABCh = 2*ABCFRQ*tCLK where tCLK is the module input clock cycle, as defined in the Section 4.2 "Module Architecture" on page 100. ABCFRQ can be programmed to values in the range of 0001000b through 1111111b. Using any other value has unpredictable results.
0	**EN (Enable).** 0: ACB is disabled, ACBCTL1, ACBST and ACBCST registers are cleared, and clocks are halted. 1: ACB is enabled.

4.8 LEGACY FUNCTIONAL BLOCKS

This section briefly describes the following blocks that provide legacy device functions:

- Parallel Port.
- Serial Port 1 and Serial Port 2 (SP1 and SP2), UART functionality for both SP1 and SP2.
- Infrared Communications Port / Serial Port 3 functionality.

Notes

- SP1 and SP2 are similar to SCC1 in the National PC87338 device.
- The Parallel Port is similar to Parallel Port in the National PC87338 device.
- The IR Communications Port is similar to SCC2 in the National PC87338 device.

The description of each Legacy block includes a general description, register maps, and bit maps. For more information about legacy blocks, contact your National Semiconductor representative.

4.8.1 Parallel Port

The Parallel Port supports all IEEE1284 standard communication modes: Compatibility (known also as Standard or SPP), Bidirectional (known also as PS/2), FIFO, EPP (known also as Mode 4) and ECP (with an optional Extended ECP mode).

4.8.1.1 Parallel Port Register and Bit Maps

The Parallel Port register maps (Table 4-33 and Table 4-34) are grouped according to first and second level offsets. EPP and second level offset registers are available only when the base address is 8-byte aligned.

Parallel Port functional block bit maps are shown in Table 4-35 and Table 4-36.

Table 4-33. Parallel Port Register Map for First Level Offset

First Level Offset	Type	Name	Modes (ECR Bits) 7 6 5
000h	R/W	**DATAR.** PP Data	000 or 001
000h	W	**AFIFO.** ECP Address FIFO	011
001h	RO	**DSR.** Status	All Modes
002h	R/W	**DCR.** Control	All Modes
003h	R/W	**ADDR.** EPP Address	100
004h	R/W	**DATA0.** EPP Data Port 0	100
005h	R/W	**DATA1.** EPP Data Port 1	100
006h	R/W	**DATA2.** EPP Data Port 2	100
007h	R/W	**DATA3.** EPP Data Port 3	100
400h	W	**CFIFO.** PP Data FIFO	010
400h	R/W	**DFIFO.** ECP Data FIFO	011
400h	R/W	**TFIFO.** Test FIFO	110
400h	RO	**CNFGA.** Configuration A	111
401h	RO	**CNFGB.** Configuration B	111
402h	R/W	**ECR.** Extended Control	All Modes
403h	R/W	**EIR.** Extended Index	All Modes
404h	R/W	**EDR.** Extended Data	All Modes
405h	R/W	**EAR.** Extended Auxiliary Status	All Modes

Table 4-34. Parallel Port Register Map for Second Level Offset

Second Level Offset	Type	Name
00h	R/W	**Control0.** Control Register 0
02h	R/W	**Control2.** Control Register 2
04h	R/W	**Control4.** Control Register 4
05h	R/W	**PP Confg0.** Parallel Port Configuration Register 0

SuperI/O Module (Continued)

Table 4-35. Parallel Port Bit Map for First Level Offset

Offset	Name	Bits							
		7	6	5	4	3	2	1	0
000h	DATAR	Data Bits							
	AFIFO	Address Bits							
001h	DSR	Printer Status	ACK# Status	PE Status	SLCT Status	ERR# Status	RSVD		EPP Timeout Status
002h	DCR	RSVD		Direction Control	Interrupt Enable	PP Input Control	Printer Initialization Control	Automatic Line Feed Control	Data Strobe Control
003h	ADDR	EPP Device or Register Selection Address Bits							
004h	DATA0	EPP Device or R/W Data							
005h	DATA1	EPP Device or R/W Data							
006h	DATA2	EPP Device or R/W Data							
007h	DATA3	EPP Device or R/W Data							
400h	CFIFO	Data Bits							
400h	DFIFO	Data Bits							
400h	TFIFO	Data Bits							
400h	CNFGA	RSVD				Bit 7 of PP Confg0	RSVD		
401h	CNFGB	RSVD	Interrupt Request Value	Interrupt Select			RSVD	DMA Channel Select	
402h	ECR	ECP Mode Control			ECP Interrupt Mask	ECP DMA Enable	ECP Interrupt Service	FIFO Full	FIFO Empty
403h	EIR	RSVD					Second Level Offset		
404h	EDR	Data Bits							
405h	EAR	FIFO Tag	RSVD						

Table 4-36. Parallel Port Bit Map for Second Level Offset

Offset	Name	Bits							
		7	6	5	4	3	2	1	0
00h	Control0	RSVD		DCR Register Live	Freeze Bit	RSVD			EPP Timeout Interrupt Mask
02h	Control2	SPP Compatibility	Channel Address Enable	RSVD	Revision 1.7 or 1.9 Select	RSVD			
04h	Control4	RSVD	PP DMA Request Inactive Time			RSVD	PP DMA Request Active Time		
05h	PP Confg0	Bit 3 of CNFGA	Demand DMA Enable	ECP IRQ Channel Number			PE Internal PU or PD	ECP DMA Channel Number	

SuperI/O Module (Continued)

4.8.2 UART Functionality (SP1 and SP2)

Both SP1 and SP2 provide UART functionality. The generic SP1 and SP2 support serial data communication with remote peripheral device or modem using a wired interface. The functional blocks can function as a standard 16450, 16550, or as an Extended UART.

4.8.2.1 UART Mode Register Bank Overview

Four register banks, each containing eight registers, control UART operation. All registers use the same 8-byte address space to indicate offsets 00h through 07h. The BSR register selects the active bank and is common to all banks. See Figure 4-18.

4.8.2.2 SP1 and SP2 Register and Bit Maps for UART Functionality

The tables in this subsection provide register and bit maps for Banks 0 through 3.

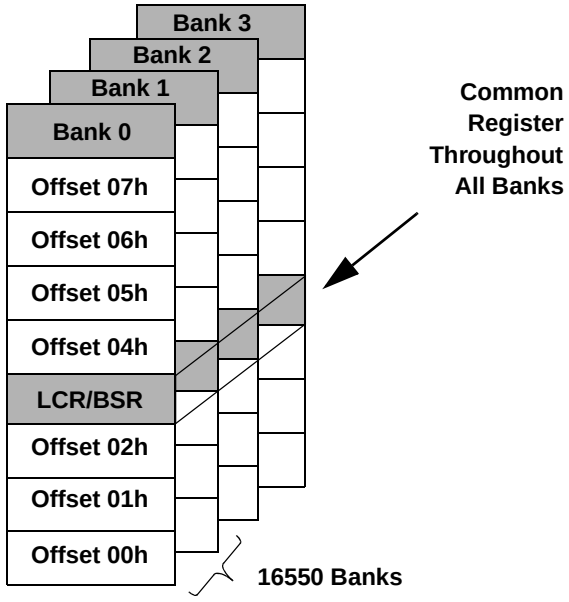


Figure 4-18. UART Mode Register Bank Architecture

Table 4-37. Bank 0 Register Map

Offset	Type	Name
00h	RO	**RXD.** Receiver Data Port
	W	**TXD.** Transmitter Data Port
01h	R/W	**IER.** Interrupt Enable
02h	RO	**EIR.** Event Identification (Read Cycles)
	R/W	**FCR.** FIFO Control (Write Cycles)
03h	W	**LCR**[1]**.** Line Control
	R/W	**BSR**[1]**.**Bank Select
04h	R/W	**MCR.** Modem/Mode Control
05h	R/W	**LSR.** Link Status
06h	R/W	**MSR.** Modem Status
07h	R/W	**SPR.** Scratchpad
	R/W	**ASCR.** Auxiliary Status and Control

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-38.

SuperI/O Module (Continued)

Table 4-38. Bank Selection Encoding

BSR Bits								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Bank Selected**
0	x	x	x	x	x	x	x	0
1	0	x	x	x	x	x	x	1
1	1	x	x	x	x	1	x	1
1	1	x	x	x	x	x	1	1
1	1	1	0	0	0	0	0	2
1	1	1	0	0	1	0	0	3

Table 4-39. Bank 1 Register Map

Offset	Type	Name
00h	R/W	**LBGD(L).** Legacy Baud Generator Divisor Port (Low Byte)
01h	R/W	**LBGD(H).** Legacy Baud Generator Divisor Port (High Byte)
02h	---	**RSVD.** Reserved
03h	W	**LCR**[1]**.** Line Control
	R/W	**BSR**[1]**.** Bank Select
04h-07h	---	**RSVD.** Reserved

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-38 on page 141.

Table 4-40. Bank 2 Register Map

Offset	Type	Name
00h	R/W	**BGD(L).** Baud Generator Divisor Port (Low Byte)
01h	R/W	**BGD(H).** Baud Generator Divisor Port (High Byte)
02h	R/W	**EXCR1.** Extended Control1
03h	R/W	**BSR.** Bank Select
04h	R/W	**EXCR2.** Extended Control 2
05h	---	**RSVD.** Reserved
06h	RO	**RXFLV.** RX_FIFO Level
07h	RO	**TXFLV.** TX_FIFO Level

Table 4-41. Bank 3 Register Map

Offset	Type	Name
00h	RO	**MRID.** Module and Revision ID
01h	RO	**SH_LCR.** Shadow of LCR
02h	RO	**SH_FCR.** Shadow of FIFO Control
03h	R/W	**BSR.** Bank Select
04h-07h	---	**RSVD.** Reserved

SuperI/O Module (Continued)

Table 4-42. Bank 0 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	RXD	RXD[7:0] (Receiver Data Bits)							
	TXD	TXD[7:0] (Transmitter Data Bits)							
01h	IER[1]	RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
	IER[2]	RSVD		TXEMP_IE	RSVD[3]/ DMA_IE[4]	MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
02h	EIR[1]	FEN[1:0]		RSVD		RXFT	IPR1	IPR0	IPF
	EIR[2]	RSVD		TXEMP_EV	RSVD [3]/ DMA_EV [4]	MS_EV	LS_EV or TXHLT_EV	TXLDL_EV	RXHDL_EV
	FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	LCR[5]	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR[5]	BKSE	BSR[6:0] (Bank Select)						
04h	MCR[1]	RSVD			LOOP	ISEN or DCDLP	RILP	RTS	DTR
	MCR[2]	RSVD				TX_DFR	RSVD	RTS	DTR
05h	LSR	ER_INF	TXEMP	TXRDY	BRK	FE	PE	OE	RXDA
06h	MSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
07h	SPR[1]	Scratch Data							
	ASCR[2]	RSVD	TXUR[4]	RXACT[4]	RXWDG[4]	RSVD	S_OET[4]	RSVD	RXF_TOUT

1. Non-Extended Mode.
2. Extended Mode.
3. In SP1 only.
4. In SP2 only.
5. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-38 on page 141.

Table 4-43. Bank 1 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	LBGD(L)	LBGD[7:0] (Low Byte)							
01h	LBGD(H)	LBGD[15:8] (High Byte)							
02h	RSVD	Reserved							
03h	LCR[1]	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR[1]	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	Reserved							

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-38 on page 141.

SuperI/O Module (Continued)

Table 4-44. Bank 2 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	BGD(L)	BGD[7:0] (Low Byte)							
01h	BGD(H)	BGD [15:8] (High Byte)							
02h	EXCR1	BTEST	RSVD	ETDLBK	LOOP	RSVD			EXT_SL
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	EXCR2	LOCK	RSVD	PRESL[1:0]		RSVD			
05h	RSVD	Reserved							
06h	RXFLV	RSVD			RFL[4:0]				
07h	TXFLV	RSVD			TFL[4:0]				

Table 4-45. Bank 3 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	MRID	MID[3:0]				RID[3:0]			
01h	SH_LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
02h	SH_FCR	RXFTH[1:0]		TXFHT[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	RSVD							

SuperI/O Module (Continued)

4.8.3 IR Communications Port (IRCP) / Serial Port 3 (SP3) Functionality

This section describes the IRCP/SP3 support registers. The IRCP/SP3 functional block provides advanced, versatile serial communications features with IR capabilities.

The IRCP/SP3 also supports two DMA channels; the functional block can use either one or both of them. One channel is required for IR-based applications, since IR communication works in half duplex fashion. Two channels would normally be needed to handle high-speed full duplex IR based applications.

The IRCP or Serial Port 3 is chosen via bit 6 of the PMR Register (see Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81).

4.8.3.1 IR/SP3 Mode Register Bank Overview

Eight register banks, each containing eight registers, control IR/SP3 operation. All registers use the same 8-byte address space to indicate offsets 00h through 07h. The BSR register selects the active bank and is common to all banks. See Figure 4-19.

4.8.3.2 IRCP/SP3 Register and Bit Maps

The tables in this subsection provide register and bit maps for Banks 0 through 7.

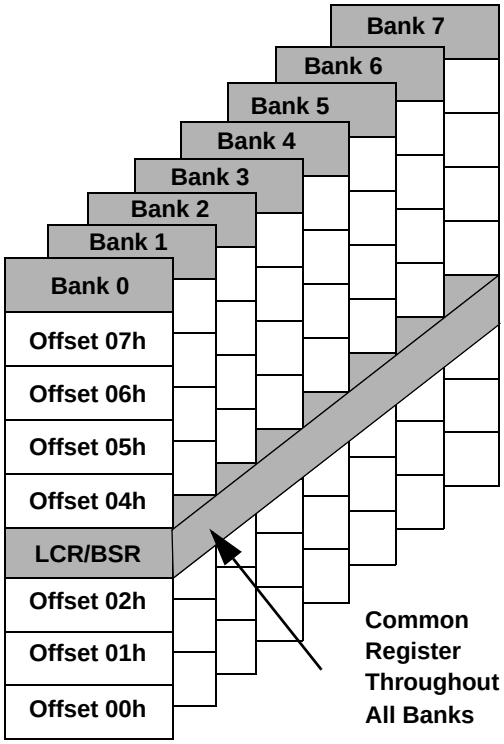


Figure 4-19. IRCP/SP3 Register Bank Architecture

Table 4-46. Bank 0 Register Map

Offset	Type	Name
00h	RO	**RXD.** Receive Data Port
	W	**TXD.** Transmit Data Port
01h	R/W	**IER.** Interrupt Enable
02h	RO	**EIR.** Event Identification
	R/W	**FCR.** FIFO Control
03h	W	**LCR**[1]**.** Link Control
	R/W	**BSR**[1]**.** Bank Select
04h	R/W	**MCR.** Modem/Mode Control
05h	R/W	**LSR.** Link Status
06h	R/W	**MSR.** Modem Status
07h	R/W	**SPR.** Scratchpad
	R/W	**ASCR.** Auxiliary Status and Control

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-47.

SuperI/O Module (Continued)

Table 4-47. Bank Selection Encoding

BSR Bits								Bank Selected	Functionality
7	6	5	4	3	2	1	0		
0	x	x	x	x	x	x	x	0	UART + IR
1	0	x	x	x	x	x	x	1	
1	1	x	x	x	x	1	x	1	
1	1	x	x	x	x	x	1	1	
1	1	1	0	0	0	0	0	2	
1	1	1	0	0	1	0	0	3	
1	1	1	0	1	0	0	0	4	IR Only
1	1	1	0	1	1	0	0	5	
1	1	1	1	0	0	0	0	6	
1	1	1	1	0	1	0	0	7	

Table 4-48. Bank 1 Register Map

Offset	Type	Name
00h	R/W	**LBGD(L).** Legacy Baud Generator Divisor Port (Low Byte)
01h	R/W	**LBGD(H).** Legacy Baud Generator Divisor Port (High Byte)
02h	---	**RSVD.** Reserved
03h	W	**LCR**[1]**.** Link Control
	R/W	**BSR**[1]**.** Bank Select
04h-07h	---	**RSVD.** Reserved

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-47.

Table 4-49. Bank 2 Register Map

Offset	Type	Name
00h	R/W	**BGD(L).** Baud Generator Divisor Port (Low Byte)
01h	R/W	**BGD(H).** Baud Generator Divisor Port (High Byte)
02h	R/W	**EXCR1.** Extended Control 1
03h	R/W	**BSR.** Bank Select
04h	R/W	**EXCR2.** Extended Control 2
05h	---	**RSVD.** Reserved
06h	RO	**TXFLV.** TX FIFO Level
07h	RO	**RXFLV.** RX FIFO Level

SuperI/O Module (Continued)

Table 4-50. Bank 3 Register Map

Offset	Type	Name
00h	RO	**MID.** Module and Revision Identification
01h	RO	**SH_LCR.** Link Control Shadow
02h	RO	**SH_FCR.** FIFO Control Shadow
03h	R/W	**BSR.** Bank Select
04h-07h	---	**RSVD.** Reserved

Table 4-51. Bank 4 Register Map

Offset	Type	Name
00h	RO	**TMR(L).** TImer (Low Byte)
01h	RO	**TMR(H).** Timer (High Byte)
02h	R/W	**IRCR1.** IR Control 1
03h	R/W	**BSR.** Bank Select
04h	R/W	**TFRL(L).** Transmission Frame Length (Low Byte)
	RO	**TFRCC(L).** Transmission Current Count (Low Byte)
05h	R/W	**TFRL(H).** Transmission Frame Length (High Byte)
	RO	**TFRCC(H).** Transmission Current Count (High Byte)
06h	R/W	**RFRML(L).** Reception Frame Maximum Length (Low Byte)
	RO	**RFRCC(L).** Reception Frame Current Count (Low Byte)
07h	R/W	**RFRML(H).** Reception Frame Maximum Length (High Byte)
	RO	**RFRCC(H).** Reception Frame Current Count (High Byte)

Table 4-52. Bank 5 Register Map

Offset	Type	Name
00h	R/W	**SPR3.** Scratchpad 2
01h	R/W	**SPR3.** Scratchpad 3
02h	R/W	**RSVD.** Reserved
03h	R/W	**BSR.** Bank Select
04h	R/W	**IRCR2.** IR Control 2
05h	RO	**FRM_ST.** Frame Status
06h	RO	**RFRL(L).** Received Frame Length (Low Byte)
	RO	**LSTFRC.** Lost Frame Count
07h	RO	**RFRL(H).** Received Frame Length (High Byte)

SuperI/O Module (Continued)

Table 4-53. Bank 6 Register Map

Offset	Type	Name
00h	R/W	**IRCR3.** IR Control 3
01h	R/W	**MIR_PW.** MIR Pulse Width
02h	R/W	**SIR_PW.** SIR Pulse Width
03h	R/W	**BSR.** Bank Select
04h	R/W	**BFPL.** Beginning Flags/Preamble Length
05h-07h	---	**RSVD.** Reserved

Table 4-54. Bank 7 Register Map

Offset	Type	Name
00h	R/W	**IRRXDC.** IR Receiver Demodulator Control
01h	R/W	**IRTXMC.** IR Transmitter Modulator Control
02h	R/W	**RCCFG.** Consumer IR (CEIR) Configuration
03h	R/W	**BSR.** Bank Select
04h	R/W	**IRCFG1.** IR Interface Configuration 1
05h-06h	---	**RSVD.** Reserved
07h	R/W	**IRCFG4.** IR Interface Configuration 4

Table 4-55. Bank 0 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	RXD	RXD[7:0] (Receive Data)							
	TXD	TXD[7:0] (Transmit Data)							
01h	IER[1]	RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
	IER[2]	TMR_IE	SFIF_IE	TXEMP_ IE/PLD_IE	DMA_IE	MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
02h	EIR[1]	FEN[1:0]		RSVD		RXFT	IPR[1:0]		IPF
	EIR[2]	TMR_EV	SFIF_EV	TXEMP_EV/ PLD_EV	DMA_EV	MS_EV	LS_EV/ TXHLT_EV	TXLDL_EV	RXHDL_EV
	FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	MCR[1]	RSVD			LOOP	ISEN/ DCDLP	RILP	RTS	DTR
	MCR[2]	MDSL[2:0]			IR_PLS	TX_DFR	DMA_EN	RTS	DTR
05h	LSR	ER_INF/ FR_END	TXEMP	TXRDY	BRK/ MAX_LEN	FE/ PHY_ERR	PE/ BAD_CRC	OE	RXDA
06h	MSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
07h	SPR[1]	Scratch Data							
	ASCR[2]	CTE/PLD	TXUR	RXACT/ RXBSY	RXWDG/ LOST_FR	TXHFE	S_EOT	FEND_INF	RXF_TOUT

1. Non-extended mode.
2. Extended mode.

SuperI/O Module (Continued)

Table 4-56. Bank 1 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	LBGD(L)	LBGD[7:0] (Low Byte Data)							
01h	LBGD(H)	LBGD[15:8] (High Byte Data)							
02h	RSVD	RSVD							
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	RSVD							

Table 4-57. Bank 2 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	BGD(L)	BGD[7:0] (Low Byte Data)							
01h	BGD(H)	BGD[15:8] (High Byte Data)							
02h	EXCR1	BTEST	RSVD	ETDLBK	LOOP	DMASWP	DMATH	DMANF	EXT_SL
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	EXCR2	LOCK	RSVD	PRESL[1:0]		RF_SIZ[1:0]		TF_SIZ[1:0]	
05h	RSVD	RSVD							
06h	TXFLV	RSVD		TFL[5:0]					
07h	RXFLV	RSVD		RFL[5:0]					

Table 4-58. Bank 3 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	MID	MID[3:0]				RID[3:0]			
01h	SH_LCR[1]	RSVD	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
02h	SH_FCR[2]	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	Reserved							

1. LCR Register Value
2. FCR Register Value

Table 4-59. Bank 4 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	TMR(L)	TMR[7:0] (Low Byte Data)							
01h	TMR(H)	RSVD				TMR[11:8] (High Byte Data)			
02h	IRCR1	RSVD				IR_SL[1:0]		CTEST	TMR_EN
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	TFRL(L)/ TFRCC(L)	TFRL[7:0] / TFRCC[7:0] (Low Byte Data)							
05h	TFRL(H)/ TFRCC(H)	RSVD				TFRL[12:8] / TFRCC[12:8] (High Byte Data)			

SuperI/O Module (Continued)

Table 4-59. Bank 4 Bit Map (Continued)

Register		Bits							
Offset	Name	7	6	5	4	3	2	1	0
06h	RFRML(L)/ RFRCC(L)	RFRML[7:0] / RFRCC[7:0] (Low Byte Data)							
07h	RFRML(H)/ RFRCC(H)	RSVD			RFRML[12:8] / RFRCC[12:8] (High Byte Data)				

Table 4-60. Bank 5 Bit Map

Register		Bits							
Offset	Name	7	6	5	4	3	2	1	0
00h	SPR2	Scratchpad 2							
01h	SPR3	Scratchpad 2							
02h	RSVD	RSVD							
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	IRCR2	RSVD	SFTSL	FEND_MD	AUX_IRRX	TX_MS	MDRS	IRMSSL	IR_FDPLX
05h	FRM_ST	VLD	LOST_FR	RSVD	MAX_LEN	PHY_ERR	BAD_CRC	OVR1	OVR2
06h	RFRL(L)/ LSTFRC	RFRL[7:0] (Low Byte Data) / LSTFRC[7:0]							
07h	RFRL(H)	RFRL[15:8] (High Byte Data)							

Table 4-61. Bank 6 Bit Map

Register		Bits							
Offset	Name	7	6	5	4	3	2	1	0
00h	IRCR3	SHDM_DS	SHMD_DS	FIR_CRC	MIR_CRC	RSVD	TXCRC_INV	TXCRC_DS	RSVD
01h	MIR_PW	RSVD				MPW[3:0]			
02h	SIR_PW	RSVD				SPW[3:0]			
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	BFPL	MBF[3:0]				FPL[3:0]			
05h-07h	RSVD	RSVD							

Table 4-62. Bank 7 Bit Map

Register		Bits							
Offset	Name	7	6	5	4	3	2	1	0
00h	IRRXDC	DBW[2:0]			DFR[4:0]				
01h	IRTXMC	MCPW[2:0]			MCFR[4:0]				
02h	RCCFG	R_LEN	T_OV	RXHSC	RCDM_DS	RSVD	TXHSC	RC_MMD[1:0]	
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	IRCFG1	STRV_MS	SIRC[2:0]			IRID3	IRIC[2:0]		
05h-06h	RSVD	RSVD							
07h	IRCFG4	RSVD	IRRX_MD	IRSL0_DS	RXINV	IRSL21_DS	RSVD		

5.0 Core Logic Module

The Core Logic module is an enhanced PCI-to-Sub-ISA bridge (South Bridge), this module is ACPI-compliant, and provides AT/Sub-ISA functionality. The Core Logic module also contains state-of-the-art power management. Two bus mastering IDE controllers are included for support of up to four ATA-compliant devices. A three-port Universal Serial Bus (USB) provides high speed, and Plug & Play expansion for a variety of new consumer peripheral devices.

5.1 FEATURE LIST

Internal Fast-PCI Interface

The internal Fast-PCI bus interface is used to connect the Core Logic and GX1 modules of the SC2200. This interface includes the following features:

- PCI protocol for transfers on Fast-PCI bus
- Up to 66 MHz operation
- Subtractive decode handled internally in conjunction with external PCI bus

Bus Mastering IDE Controllers

- Two controllers with support for up to four IDE devices
- Independent timing for master and slave devices for both channels
- PCI bus master burst reads and writes
- Multiword DMA support
- Programmed I/O (PIO) Modes 0-4 support

Universal Serial Bus

- Three independent USB interfaces
- Open Host Controller Interface (OpenHCI) specification compliant

PCI Interface

- PCI 2.1 compliant
- PCI master for AC97 and IDE controllers
- Subtractive agent for unclaimed transactions
- Supports PCI initiator-to-Sub-ISA cycle translations
- PCI-to-Sub-ISA interrupt mapper/translator
- External PCI bus
 - — Devices internal to the Core Logic module (IDE, Audio, USB, Sub-ISA, etc.) cannot master to memory through the external PCI bus.
 - — Legacy DMA is not supported to memory located on external PCI bus.
 - — The Core Logic module does not transfer subtractively decoded I/O cycles originating from the external PCI bus.

AT Compatibility

- 8259A-equivalent interrupt controllers
- 8254-equivalent timer
- 8237-equivalent DMA controllers
- Port A, B, and NMI logic
- Positive decode for AT I/O space

Sub-ISA Interface

- Boot ROM chip select
- Extended ROM to 16 MB
- Two general-purpose chip selects
- NAND Flash support
- M-Systems DiskOnChip support
- Is not the subtractive decode agent

Power Management

- Automated CPU 0V Suspend modulation
- I/O Traps and Idle Timers for peripheral power management
- Software SMI and Stop Clock for APM support
- ACPI-compliant timer and register set
- Up to 22 GPIOs of which all can generate Power Management Interrupts (PMEs)
- Three Dedicated GPWIOs powered by V_{SBL} and V_{SB}
- Shadow register support for legacy controllers for 0V Suspend

Integrated Audio

- AC97 Version 2.0 compliant interface to audio codecs
- Secondary codec support
- AMC97 codec support

Video Processor Interface

- Synchronous serial interface to the Video Processor
- Translates video and clock control register accesses from PCI to serial interface
- Supports both reads and writes of Video Processor registers
- Retries Fast-PCI bus accesses until Core Logic completes the transfer over the serial interface

Low Pin Count (LPC) Interface

- Based on Intel LPC Interface Specification Revision 1.0
- Serial IRQ support

Core Logic Module (Continued)

5.2 MODULE ARCHITECTURE

The Core Logic architecture provides the internal functional blocks shown in Figure 5-1.

- Fast-PCI interface to external PCI bus
- IDE controllers (UDMA-33)
- USB controllers
- Sub-ISA bus interface
- AT compatibility logic (legacy)
- ACPI compliant power management (includes GPIO interfaces, such as joystick)
- Integrated audio controller
- Low Pin Count (LPC) Interface

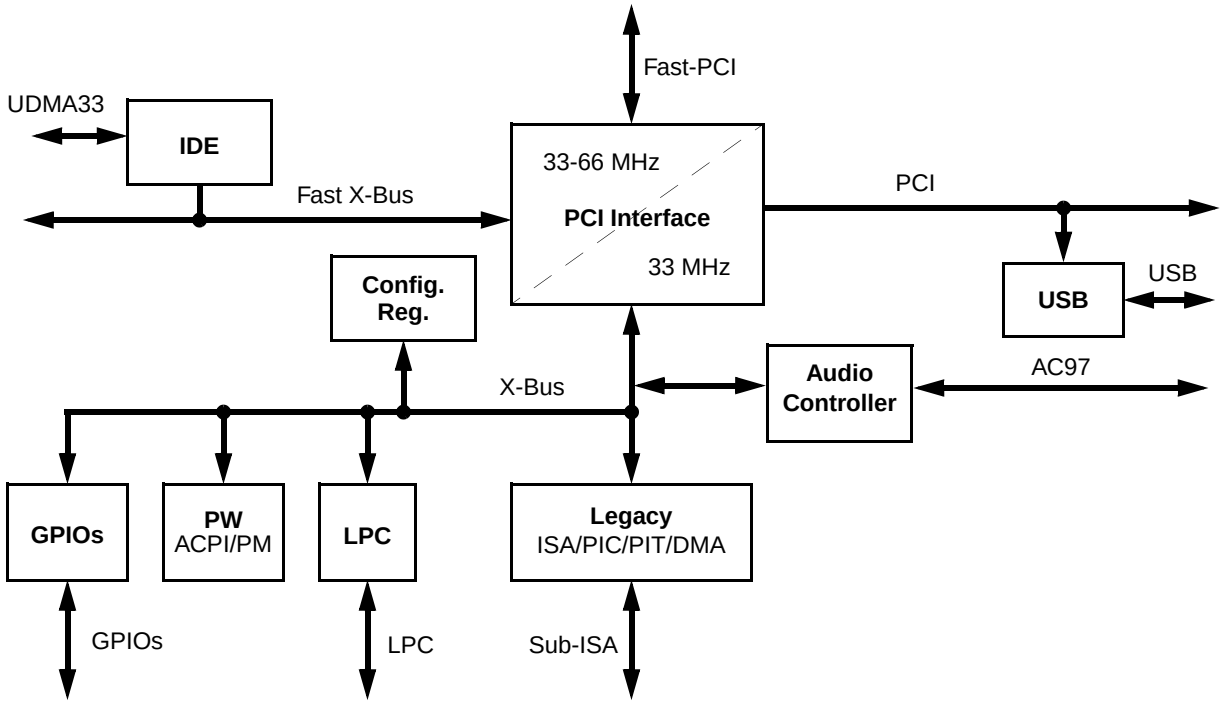


Figure 5-1. Core Logic Module Block Diagram

Core Logic Module (Continued)

5.2.1 Fast-PCI Interface to External PCI Bus

The Core Logic module provides a PCI bus interface that is both a slave for PCI cycles initiated by the GX1 module or other PCI master devices, and a non-preemptive master for DMA transfer cycles. It is also a standard PCI master for the IDE controllers and audio I/O logic. The Core Logic supports positive decode for configurable memory and I/O regions, and implements a subtractive decode option for unclaimed PCI accesses. It also generates address and data parity, and performs parity checking. The arbiter for the Fast-PCI interface is located in the GX1 module.

Configuration registers are accessed through the PCI interface using the PCI Bus Type 1 configuration mechanism as described in the PCI Specification.

5.2.1.1 Processor Mastered Cycles

The Core Logic module acts on all processor initiated cycles according to PCI rules for active/subtractive decode using DEVSEL#. Memory writes are automatically posted. Reads are retried if they are *not* destined for actively decoded (i.e., positive decode) devices on the high speed X-Bus or the 33 MHz X-Bus. This means that reads to external PCI, LPC, or Sub-ISA devices are automatically treated as delayed transactions through the PCI retry mechanism. This allows the high bandwidth devices access to the Fast-PCI interface while the response from a slow device is accumulated.

Bursting from the host is not supported.

All types of configuration cycles are supported and handled appropriately according to the PCI specification.

5.2.1.2 External PCI Mastered Cycles

Memory cycles mastered by external PCI devices on the external PCI bus are actively taken if they are to the system memory address range. Memory cycles to system memory are forwarded to the Fast-PCI interface. Burst transfers are stopped on every cache line boundary to allow efficient buffering in the Fast-PCI interface block.

I/O and configuration cycles mastered by external PCI devices which are subtractively decoded by the Core Logic module, are not handled.

5.2.1.3 Core Logic Internal or Sub-ISA Mastered Cycles

Only memory cycles (not I/O cycles) are supported by the internal Sub-ISA or legacy DMA masters. These memory cycles are always forwarded to the Fast-PCI interface.

5.2.1.4 External PCI Bus

The external PCI bus is a fully-compliant PCI bus. PCI slots are connected to this bus. Support for up to two bus masters is provided. The arbiter is in the Core Logic module.

5.2.1.5 Bus Master Request Priority

The Fast-PCI bus supports seven bus masters. The requests (REQs) are fixed in priority. The seven bus masters in order of priority are:

1) VIP
2) IDE Channel 0
3) IDE Channel 1
4) Audio
5) USB
6) External REQ0#
7) External REQ1#

5.2.2 PSERIAL Interface

The majority of the system power management logic is implemented in the Core Logic module, but a minimal amount of logic is contained within the GX1 module to provide information that is not externally visible (e.g., graphics controller).

The GX1 module implements a simple serial communications mechanism to transmit the CPU status to the Core Logic module via internal signal PSERIAL. The GX1 module accumulates CPU events in an 8-bit register which it transmits serially every 1 to 10 µs.

The packet transmitter holds the serial output internal signal (PSERIAL) low until the transmission interval counter has elapsed. Once the counter has elapsed, the PSERIAL signal is held high for two clocks to indicate the start of packet transmission. The contents of the Serial Packet register are then shifted out starting from bit 7 down to bit 0. The PSERIAL signal is held high for one clock to indicate the end of packet transmission and then remains low until the next transmission interval. After the packet transmission is complete, the GX1 module's Serial Packet register's contents are cleared.

The GX1 module's input clock is used as the clock reference for the serial packet transmitter.

Once a bit in the register is set, it remains set until the completion of the next packet transmission. Successive events of the same type that occur between packet transmissions are ignored. Multiple unique events between packet transmissions accumulate in this register. The GX1 module transmits the contents of the serial packet only when a bit in the Serial Packet register is set and the interval counter has elapsed.

The Core Logic module decodes the serial packet after each transmission and performs the power management tasks related to video retrace.

For more information on the Serial Packet register refer to the *GX1 Processor Series Datasheet*.

5.2.2.1 Video Retrace Interrupt

Bit 7 of the "Serial Packet" can be used to generate an SMI whenever a video retrace occurs within the GX1 module. This function is normally not used for power management but for SoftVGA routines. Setting F0 Index 83h[2] = 1 enables this function. A read only status register located at F1BAR0+I/O Offset 00h[5] can be read to see if the SMI was caused by a video retrace event.

Core Logic Module (Continued)

5.2.3 IDE Controller

The Core Logic module integrates a PCI bus mastering, ATA-4 compatible IDE controller. This controller supports UltraDMA, Multiword DMA and Programmed I/O (PIO) modes. Two devices are supported on the IDE controller. The data-transfer speed for each device can be independently programmed. This allows high-speed IDE peripherals to coexist on the same channel as lower speed devices.

The Core Logic module supports two IDE channels, a primary channel and a secondary channel.

The IDE interface provides a variety of features to optimize system performance, including 32-bit disk access, post write buffers, bus master, Multiword DMA, look-ahead read buffer, and prefetch mechanism for each channel respectively.

The IDE interface timing is completely programmable. Timing control covers the command active and recover pulse widths, and command block register accesses. The IDE data-transfer speed for each device on each channel can be independently programmed allowing high-speed IDE peripherals to coexist on the same channel as older, compatible devices.

The Core Logic module also provides a software accessible buffered reset signal to the IDE drive, F0 Index 44h[3:2]. The IDE_RST# signal is driven low during reset to the Core Logic module and can be driven low or high as needed for device-power-off conditions.

5.2.3.1 IDE Configuration Registers

Registers for configuring Channels 0 and 1 are located in the PCI register space designated as Function 2 (F2 Index 40h-5Ch). Table 5-35 on page 263 provides the bit formats for these registers. The IDE bus master configuration registers are accessed via F2 Index 20h which is Base Address Register 4 in Function 2 (F2BAR4). See Table 5-36 on page 267 for register/bit formats.

The following subsections discuss Core Logic operational/programming details concerning PIO, Bus Master, and UltraDMA/33 modes.

5.2.3.2 PIO Mode

The IDE data port transaction latency consists of address latency, asserted latency and recovery latency. Address latency occurs when a PCI master cycle targeting the IDE data port is decoded, and the IDE_ADDR[2:0] and IDE_CS# lines are not set up. Address latency provides the setup time for the IDE_ADDR[2:0] and IDE_CS# lines prior to IDE_IOR# and IDE_IOW#.

Asserted latency consists of the I/O command strobe assertion length and recovery time. Recovery time is provided so that transactions may occur back-to-back on the IDE interface without violating minimum cycle periods for the IDE interface.

If IDE_IORDY is asserted when the initial sample point is reached, no wait states are added to the command strobe assertion length. If IDE_IORDY is negated when the initial sample point is reached, additional wait states are added.

Recovery latency occurs after the IDE data port transactions have completed. It provides hold time on the IDE_ADDR[2:0] and IDE_CS# lines with respect to the read and write strobes (IDE_IOR# and IDE_IOW#).

The PIO portion of the IDE registers is enabled through:

- Channel 0 Drive 0 Programmed I/O Register (F2 Index 40h)
- Channel 0 Drive 1 Programmed I/O Register (F2 Index 48h)
- Channel 1 Drive 0 Programmed I/O Register (F2 Index 50h)
- Channel 1 Drive 1 Programmed I/O Register (F2 Index 58h)

The IDE channels and devices can be individually programmed to select the proper address setup time, asserted time, and recovery time.

The bit formats for these registers are shown in Table 5-35 on page 263. Note that there are different bit formats for each of the PIO programming registers depending on the operating format selected: Format 0 or Format 1:

- F2 Index 44h[31] (Channel 0 Drive 0 — DMA Control Register) sets the format of the PIO register.
 - — If bit 31 = 0, Format 0 is used and it selects the slowest PIO mode (bits [19:16]) per channel for commands.
 - — If bit 31 = 1, Format 1 is used and it allows independent control of command and data.

Also listed in the bit formats are recommended values for the different PIO modes. Note that these are only recommended settings and are not 100% tested.

When using independent control of command and data cycles the following algorithm should be used when two IDE devices are sharing the same channel:

1) The PIO data cycle timing for a particular device can be the timing value for the maximum PIO mode which that device reports it supports.
2) The PIO command cycle timing for a particular device must be the timing value for the lowest PIO mode for both devices on the channel.

For example, if a channel had one Mode 4 device and one Mode 0 device, then the Mode 4 device would have command timings for Mode 0 and data timing for Mode 4. The Mode 0 device would have both command and data timings for Mode 0. Note that for the Mode 0 case, the 32-bit timing value is listed because both data and command timings are the same mode. However, the actual timing value for the Mode 4 device would be constructed out of the Mode 4 data timing 16-bit value and the Mode 0 16-bit command timing value. Both 16-bit values are shown in the register description but not assembled together as they are mixed modes.

Core Logic Module (Continued)

5.2.3.3 Bus Master Mode

Two IDE bus masters are provided to perform the data transfers for the primary and secondary channels. The IDE controller of the Core Logic module off-loads the CPU and improves system performance in multitasking environments.

The bus master mode programming interface is an extension of the standard IDE programming model. This means that devices can always be dealt with using the standard IDE programming model, with the master mode functionality used when the appropriate driver and devices are present. Master operation is designed to work with any IDE device that supports DMA transfers on the IDE bus. Devices that work in PIO mode can only use the standard IDE programming model.

The IDE bus masters use a simple scatter/gather mechanism allowing large transfer blocks to be scattered to or gathered from memory. This cuts down on the number of interrupts to and interactions with the CPU.

Physical Region Descriptor Table Address

Before the controller starts a master transfer it is given a pointer to a Physical Region Descriptor Table. This pointer sets the starting memory location of the Physical Region Descriptors (PRDs). The PRDs describe the areas of memory that are used in the data transfer. The PRDs must be aligned on a 4-byte boundary and the table cannot cross a 64 KB boundary in memory.

Primary and Secondary IDE Bus Master Registers

The IDE Bus Master Registers for each channel (primary and secondary) have an IDE Bus Master Command register and Bus Master Status register. These registers and bit formats are described in Table 5-36 on page 267.

Physical Region Descriptor Format

Each physical memory region to be transferred is described by a Physical Region Descriptor (PRD) as illustrated in Table 5-1. When the bus master is enabled (Command register bit 0 = 1), data transfer proceeds until each PRD in the PRD table has been transferred. The bus master does not cache PRDs.

The PRD table consists of two DWORDs. The first DWORD contains a 32-bit pointer to a buffer to be transferred. The second DWORD contains the size (16 bits) of the buffer and the EOT flag. The EOT bit (bit 31) must be set to indicate the last PRD in the PRD table.

Programming Model

The following steps explain how to initiate and maintain a bus master transfer between memory and an IDE device.

1) Software creates a PRD table in system memory. Each PRD entry is 8 bytes long, consisting of a base address pointer and buffer size. The maximum data that can be transferred from a PRD entry is 64 KB. A PRD table must be aligned on a 4-byte boundary. The last PRD in a PRD table must have the EOT bit set.
2) Software loads the starting address of the PRD table by programming the PRD Table Address register.
3) Software must fill the buffers pointed to by the PRDs with IDE data.
4) Write 1 to the Bus Master Interrupt bit and Bus Master Error (Status register bits 2 and 1) to clear the bits.
5) Set the correct direction to the Read or Write Control bit (Command register bit 3).

 Engage the bus master by writing a “1” to the Bus Master Control bit (Command register bit 0).

 The bus master reads the PRD entry pointed to by the PRD Table Address register and increments the address by 08h to point to the next PRD. The transfer begins.
6) The bus master transfers data to/from memory responding to bus master requests from the IDE device. At the completion of each PRD, the bus master’s next response depends on the settings of the EOT flag in the PRD. If the EOT bit is set, then the IDE bus master clears the Bus Master Active bit (Status register bit 0) and stop. If any errors occurred during the transfer, the bus master sets the Bus Master Error bit Status register bit 1).

Table 5-1. Physical Region Descriptor Format

	Byte 3								Byte 2								Byte 1								Byte 0							
DWORD	**31**	**31**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	Memory Region Physical Base Address [31:1] (IDE Data Buffer)																															0
1	E O T	Reserved															Size [15:1]															0

Core Logic Module (Continued)

5.2.3.4 UltraDMA/33 Mode

The IDE controller of the Core Logic module supports UltraDMA/33. It utilizes the standard IDE Bus Master functionality to interface, initiate and control the transfer. UltraDMA/33 definition also incorporates a Cyclic Redundancy Checking (CRC) error checking protocol to detect errors.

The UltraDMA/33 protocol requires no extra signal pins on the IDE connector. The IDE controller redefines three standard IDE control signals when in UltraDMA/33 mode. These definitions are shown in Table 5-2.

Table 5-2. UltraDMA/33 Signal Definitions

IDE Controller Channel Signal	UltraDMA/33 Read Cycle	UltraDMA/33 Write Cycle
IDE_IOW#	STOP	STOP
IDE_IOR#	DMARDY#	STROBE
IDE_IORDY	STROBE	DMARDY#

All other signals on the IDE connector retain their functional definitions during the UltraDMA/33 operation.

IDE_IOW# is defined as STOP for both read and write transfers to request to stop a transaction.

IDE_IOR# is redefined as DMARDY# for transferring data from the IDE device to the IDE controller. It is used by the IDE controller to signal when it is ready to transfer data and to add wait states to the current transaction. IDE_IOR# signal is defined as STROBE for transferring data from the IDE controller to the IDE device. It is the data strobe signal driven by the IDE controller on which data is transferred during each rising and falling edge transition.

IDE_IORDY is redefined as STROBE for transferring data from the IDE device to the IDE controller during a read cycle. It is the data strobe signal driven by the IDE device on which data is transferred during each rising and falling edge transition. IDE_IORDY is defined as DMARDY# during a write cycle for transferring data from the IDE controller to the IDE device. It is used by the IDE device to signal when it is ready to transfer data and to add wait states to the current transaction.

UltraDMA/33 data transfer consists of three phases, a startup phase, a data transfer phase and a burst termination phase.

The IDE device begins the startup phase by asserting IDE_DREQ. When ready to begin the transfer, the IDE controller asserts IDE_DACK#. When IDE_DACK# is asserted, the IDE controller drives IDE_CS0# and IDE_CS1# asserted, and IDE_ADDR[2:0] low. For write cycles, the IDE controller negates STOP, waits for the IDE device to assert DMARDY#, and then drives the first data WORD and STROBE signal. For read cycles, the IDE controller negates STOP, and asserts DMARDY#. The IDE device then sends the first data WORD and asserts STROBE.

The data transfer phase continues the burst transfers with the Core Logic and the IDE via providing data, toggling STROBE and DMARDY#. The IDE_DATA[15:0] is latched by receiver on each rising and falling edge of STROBE. The transmitter can pause the burst cycle by holding STROBE high or low, and resume the burst cycle by again toggling STROBE. The receiver can pause the burst cycle by negating DMARDY# and resumes the burst cycle by asserting DMARDY#.

The current burst cycle can be terminated by either the transmitter or the receiver. A burst cycle must first be paused as described above before it can be terminated. The IDE controller can then stop the burst cycle by asserting STOP, with the IDE device acknowledging by negating IDE_DREQ. The IDE device then stops the burst cycle by negating IDE_DREQ and the IDE controller acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The IDE controller then puts the result of the CRC calculation onto the IDE_DATA[15:0] while de-asserting IDE_DACK#. The IDE device latches the CRC value on the rising edge of IDE_DACK#.

The CRC value is used for error checking on UltraDMA/33 transfers. The CRC value is calculated for all data by both the IDE controller and the IDE device during the UltraDMA/33 burst transfer cycles. This result of the CRC calculation is defined as all data transferred with a valid STROBE edge while IDE_DACK# is asserted. At the end of the burst transfer, the IDE controller drives the result of the CRC calculation onto IDE_DATA[15:0] which is then strobed by the de-assertion of IDE_DACK#. The IDE device compares the CRC result of the IDE controller to its own and reports an error if there is a mismatch.

The timings for UltraDMA/33 are programmed into the DMA control registers:

- Channel 0 Drive 0 DMA Control Register (F2 Index 44h)
- Channel 0 Drive 1 DMA Control Register (F2 Index 4Ch)
- Channel 1 Drive 0 DMA Control Register (F2 Index 54h)
- Channel 1 Drive 1 DMA Control Register (F2 Index 5Ch)

The bit formats for these registers are described in Table 5-35 on page 263. Note that F2 Index 44h[20] is used to select either Multiword or UltraDMA mode. Bit 20 = 0 selects Multiword DMA mode. If bit 20 = 1, then UltraDMA/33 mode is selected. Once mode selection is made using this bit, the remaining DMA Control registers also operate in the selected mode.

Also listed in the bit formats are recommended values for both Multiword DMA Modes 0-2 and UltraDMA/33 Modes 0-2. Note that these are only recommended settings and are not 100% tested.

Core Logic Module (Continued)

5.2.4 Universal Serial Bus

The Core Logic module provides three complete, independent USB ports. Each port has a Data "Negative" and a Data "Positive" signal.

The USB ports are Open Host Controller Interface (OpenHCI) compliant. The OpenHCI specification provides a register-level description for a host controller, as well as common industry hardware/software interface and drivers.

5.2.5 Sub-ISA Bus Interface

The Sub-ISA interface of the Core Logic module is an ISA-like bus interface that is used by SC2200 to interface with Boot Flash, M-Systems DiskOnChip or NAND EEPROM and other I/O devices. The Core Logic module is the default subtractive decoding agent and forwards all unclaimed memory and I/O cycles to the ISA bus. However, the Core Logic module can be configured to ignore either I/O, memory, or all unclaimed cycles (subtractive decode disabled).

Note: The external Sub-ISA bus is a positive decode bus. Unclaimed memory and I/O cycles will not appear on the Sub-ISA interface.

The Core Logic module does not support Sub-ISA refresh cycles. The refresh toggle bit in Port B still exists for software compatibility reasons.

The Sub-ISA interface includes the followings signals in addition to the signals used for an ISA interface:

- IOCS0#/IOCS1#
 - — Asserted on I/O read/write transactions from/to a programmable address range.
- DOCCS#
 - — Asserted on memory read/write transactions from/to a programmable window.
- ROMCS#
 - — Asserted on memory read/write to upper 16 MB of address space. Configurable via the ROM Mask register (F0 Index 6Eh).
- DOCR#
 - — DOCR# is asserted on memory read transactions from DOCCS# window (i.e., when both DOCCS# and MEMR# are active, DOCR# is active; otherwise, it is inactive).
- DOCW
 - — DOCW# is asserted on memory write transactions to DOCCS# window (i.e., when both DOCCS# and MEMW# are active, DOCW# is active; otherwise, it is inactive).
- RD#, WR#
 - — The signals IOR#, IOW#, MEMR#, and MEMW# are combined into two signals: RD# is asserted on I/O read or memory read; WR# is asserted on I/O write or memory write.

Memory devices that use ROMCS# or DOCCS# as their chip select signal can be configured to support an 8-bit or 16-bit data bus via bits 3 and 6 of the MCR register. Such devices can also be configured as zero wait states devices (regardless of the data bus width) via bits 9 and 10 of the MCR register. For MCR register bit descriptions, see Table 3-2 on page 81.

I/O peripherals that use IOCS0# or IOCS1# as their chip select signal can be configured to support an 8-bit or 16-bit data bus via bits 7 and 8 of the MCR register. Such devices can also be configured as zero wait state devices (for 8-bit peripherals) via bits 11 and 12 of the MCR register. For MCR register bit descriptions, see Table 3-2 on page 81.

Other memory devices and I/O peripherals must be 8-bit devices; their transactions can not be with zero wait states.

The Boot Flash supported by the SC2200 can be up to 16 MB. It is supported with the ROMCS# signal.

All unclaimed memory and I/O cycles are forwarded to the Internal ISA bus if subtractive decode is enabled.

The DiskOnChip chip select signal (DOCCS#) is asserted on any memory read or memory write transaction from/to a programmable address range. The address range is programmable via the DOCCS# Base Address and Control registers (F0 Index 78h and 7Ch). The base address must be on an address boundary, the size of the range.

Signal DOCCS# can also be used to interface to NAND Flash devices together with signals DOCW# and DOCR#. See application note *Geode™ SC1200/SC2200/SC3200 IAOC Devices: External NAND Flash Memory Circuit* for details.

Core Logic Module (Continued)

5.2.5.1 Sub-ISA Bus Cycles

The ISA bus controller issues multiple ISA cycles to satisfy PCI transactions that are larger than 16 bits. A full 32-bit read or write results in two 16-bit ISA transactions or four 8-bit ISA transactions. The ISA controller gathers the data from multiple ISA read cycles and returns TRDY# to the PCI bus.

SA[23:0] are a concatenation of ISA LA[23:17] and SA[19:0] and perform equivalent functionality at a reduced pin count.

Figure 5-2 shows the relationship between a PCI cycle and the corresponding ISA cycle generated.

Note: Not all signals described in Figure 5-2 are available externally. See Section 2.4.7 "Sub-ISA Interface Signals" on page 66 for more information about which Sub-ISA signals are externally available on the SC2200.

5.2.5.2 Sub-ISA Support of Delayed PCI Transactions

Multiple PCI cycles occur for every slower ISA cycle. This prevents slow PCI cycles from occupying too much bandwidth and allows access to other PCI traffic. Figure 5-3 on page 158 shows the relationship of PCI cycles to an ISA cycle with PCI delayed transactions enabled.

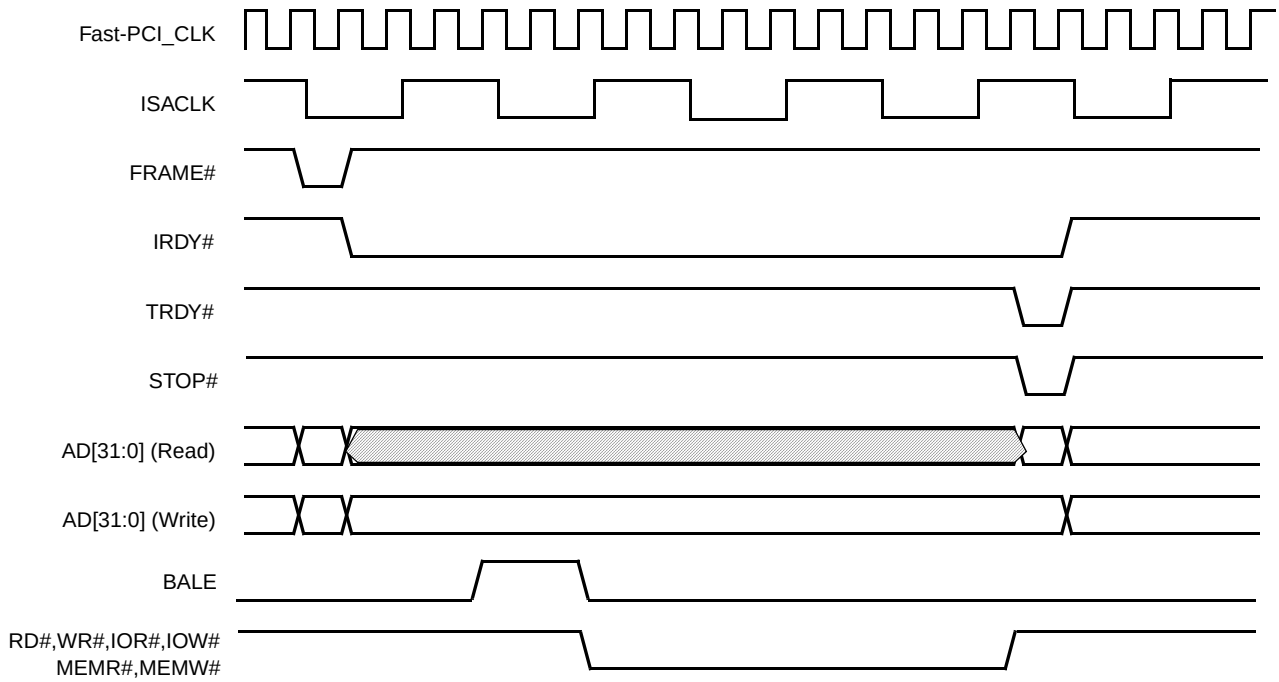


Figure 5-2. Non-Posted Fast-PCI to ISA Access

Core Logic Module (Continued)

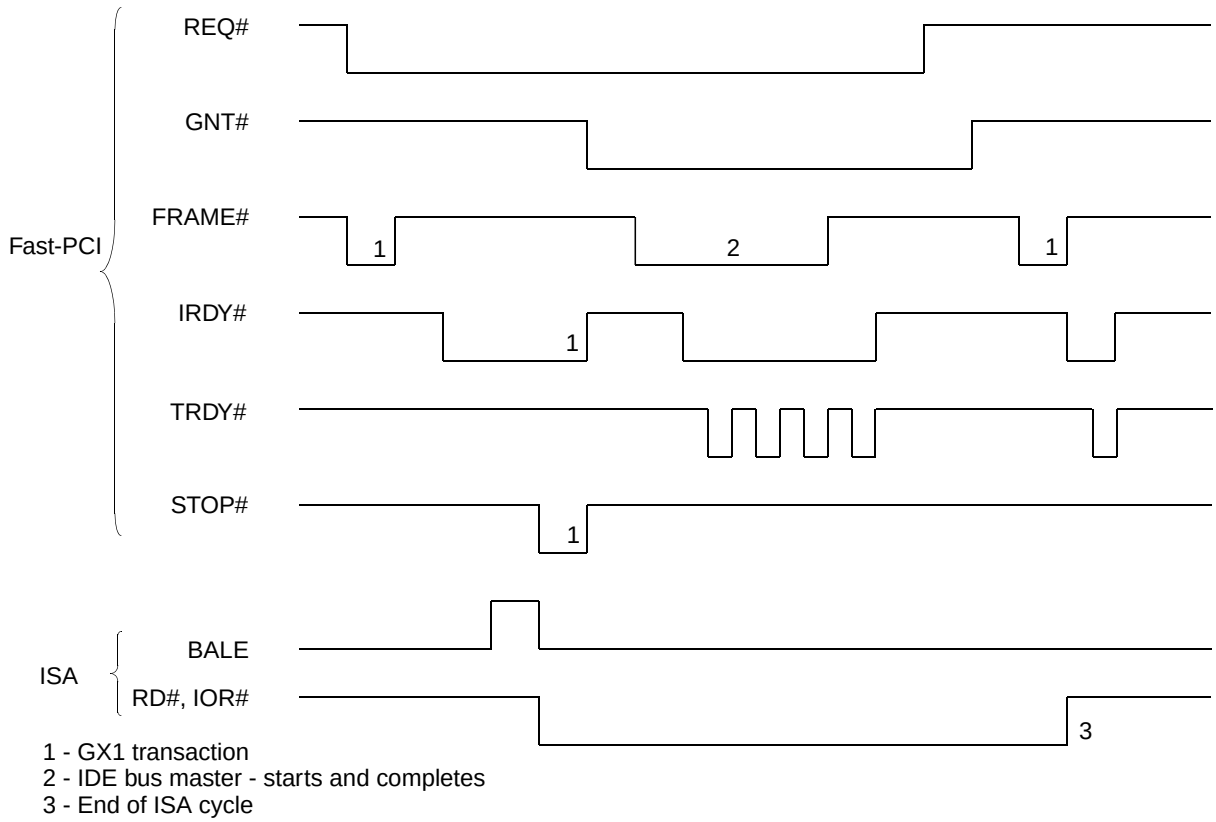


Figure 5-3. PCI to ISA Cycles with Delayed Transaction Enabled

5.2.5.3 Sub-ISA Bus Data Steering

The Core Logic module performs all of the required data steering from SD[7:0] to SD[15:0] during normal 8-bit ISA cycles, as well as during DMA and ISA master cycles. It handles data transfers between the 32-bit PCI data bus and the ISA bus. 8/16-bit devices can reside on the ISA bus. Various PC-compatible I/O registers, DMA controller registers, interrupt controller registers, and counter/timer registers lie on the on-chip I/O data bus. Either the PCI bus master or the DMA controllers can become the bus owner.

When the PCI bus master is the bus owner, the Core Logic module data steering logic provides data conversion necessary for 8/16/32-bit transfers to and from 8/16-bit devices on either the Sub-ISA bus or the 8-bit registers on the on-chip I/O data bus. When PCI data bus drivers of the Core Logic module are in TRI-STATE, data transfers between the PCI bus master and PCI bus devices are handled directly via the PCI data bus.

When the DMA requestor is the bus owner, the Core Logic module allows 8/16-bit data transfer between the Sub-ISA bus and the PCI data bus.

5.2.5.4 I/O Recovery Delays

In normal operation, the Core Logic module inserts a delay between back-to-back ISA I/O cycles that originate on the PCI bus. The default delay is four ISACLK cycles. Thus, the second of consecutive I/O cycles is held in the ISA bus controller until this delay count has expired. The delay is measured between the rising edge of IOR#/IOW# and the falling edge of BALE. This delay can be adjusted to a greater delay through the ISA I/O Recovery Control register (F0 Index 51h).

Note: This delay is not inserted for a 16-bit Sub-ISA I/O access that is split into two 8-bit I/O accesses.

Core Logic Module (Continued)

5.2.5.5 ISA DMA

DMA transfers occur between ISA I/O peripherals and system memory (i.e., not available externally). The data width can be either 8 or 16 bits. Out of the seven DMA channels available, four are used for 8-bit transfers while the remaining three are used for 16-bit transfers. One byte or WORD is transferred in each DMA cycle.

Note: The Core Logic module does not support DMA transfers to ISA memory.

The ISA DMA device initiates a DMA request by asserting one of the DRQ[7:5, 3:0] signals. When the Core Logic module receives this request, it sends a bus grant request to the PCI arbiter. After the PCI bus has been granted, the respective DACK# is driven active.

The Core Logic module generates PCI memory read or write cycles in response to a DMA cycle. Figure 5-4 and Figure 5-5 are examples of DMA memory read and memory write cycles. Upon detection of the DMA controller's MEMR# or MEMW# active, the Core Logic module starts the PCI cycle, asserts FRAME#, and negates an internal IOCHRDY. This assures the DMA cycle does not complete before the PCI cycle has provided or accepted the data. IOCHRDY is internally asserted when IRDY# and TRDY# are sampled active.

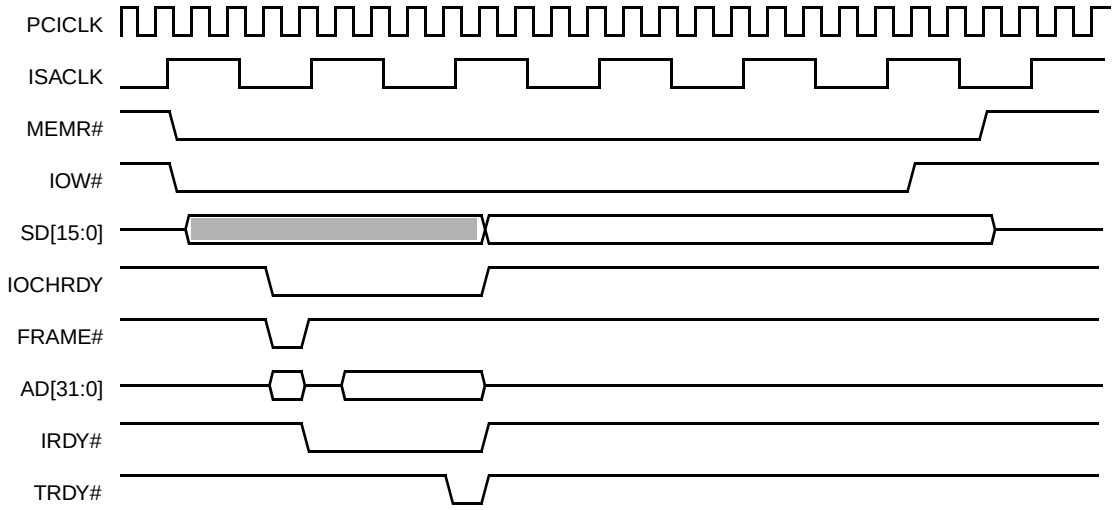


Figure 5-4. ISA DMA Read from PCI Memory

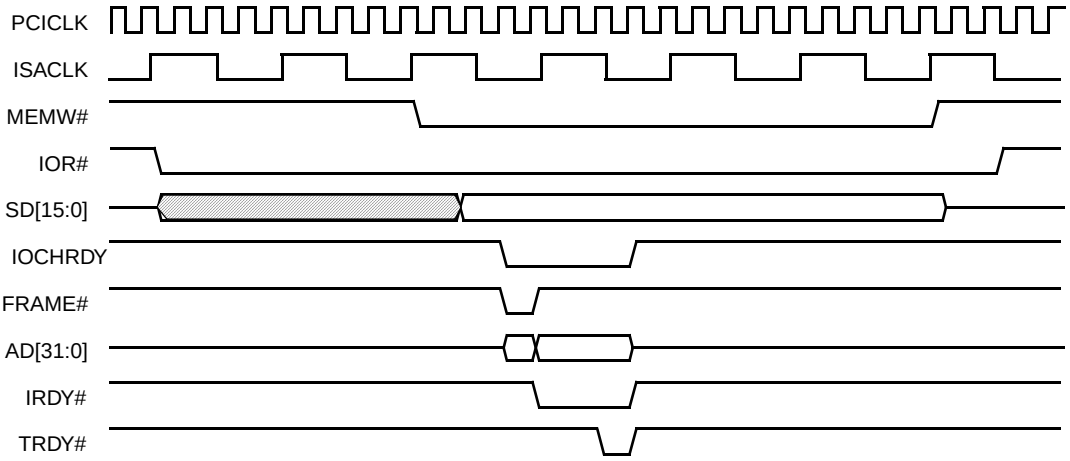


Figure 5-5. ISA DMA Write to PCI Memory

Core Logic Module (Continued)

5.2.5.6 ROM Interface

The Core Logic module positively decodes memory addresses 000F0000h-000FFFFFh (64 KB) and FFFC0000h-FFFFFFFFh (256 KB) at reset. These memory cycles cause the Core Logic module to claim the cycle, and generate an ISA bus memory cycle with ROMCS# asserted. The Core Logic module can also be configured to respond to memory addresses FF000000h-FFFFFFFFh (16 MB) and 000E0000h-000FFFFFh (128 KB).

8- or 16-bit wide ROM is supported. BOOT16 strap determines the width after reset. MCR[14,3] (Offset 34h) in the General Configuration Block (see Table 3-2 on page 81 for bit details) allows program control of the width.

Flash ROM is supported in the Core Logic module by enabling the ROMCS# signal on write accesses to the ROM region. Normally only read cycles are passed to the ISA bus, and the ROMCS# signal is suppressed for write cycles. When the ROM Write Enable bit (F0 Index 52h[1]) is set, a write access to the ROM address region causes a write cycle to occur with MEMW#, WR# and ROMCS# asserted.

5.2.5.7 PCI and Sub-ISA Signal Cycle Multiplexing

The SC2200 multiplexes most PCI and Sub-ISA signals on the balls listed in Table 5-3, in order to reduce the number of balls on the device. Cycle multiplexing is on a bus-cycle by bus-cycle basis (see Figure 5-6 on page 161), where the internal Core Logic PCI bridge arbitrates between PCI cycles and Sub-ISA cycles. Other PCI and Sub-ISA signals remain non-shared, however, some Sub-ISA signals may be muxed with GPIO.

Sub-ISA cycles are only generated as a result of GX1 module accesses to the following addresses or conditions:

- ROMCS# address range.
- DOCCS# address range.
- IOCS0# address range.
- IOCS1# address range.
- An I/O write to address 80h or to 84h.
- Internal ISA is programmed to be the subtractive decode agent and no other agents claim the cycle.

If the Sub-ISA and PCI bus have more than four components, the Sub-ISA components can be buffered using 74HCT245 or 74FCT245 type transceivers. The RD# (an AND of IOR#, MEMR#) signal can be used as DIR control while TRDE# is used as enable control.

Table 5-3. Cycle Multiplexed PCI / Sub-ISA Balls

PCI	Sub-ISA	Ball No.	
		EBGA	TEPBGA
AD0	A0	A17	U1
AD1	A1	D16	P3
AD2	A2	A18	U3
AD3	A3	A15	N1
AD4	A4	A16	P1
AD5	A5	A14	N3
AD6	A6	C15	N2
AD7	A7	B14	M2
AD8	A8	C14	M4
AD9	A9	B13	L2
AD10	A10	C13	L3
AD11	A11	C12	K1
AD12	A12	A12	L4
AD13	A13	C11	J1
AD14	A14	A11	K4
AD15	A15	B10	J3
AD16	A16	A7	E1
AD17	A17	C7	F4
AD18	A18	D7	E3
AD19	A19	A6	E2
AD20	A20	D6	D3
AD21	A21	C6	D1
AD22	A22	A5	D2
AD23	A23	F4	B6
AD24	D0	C5	C2
AD25	D1	D5	C4
AD26	D2	A4	C1
AD27	D3	B4	D4
AD28	D4	C4	B4
AD29	D5	A3	B3
AD30	D6	C2	A3
AD31	D7	B3	D5
C/BE0#	D8	A13	L1
C/BE1#	D9	A10	J2
C/BE2#	D10	D8	F3
C/BE3#	D11	A8	H4
PAR	D12	C10	J4
TRDY#	D13	B8	F1
IRDY#	D14	C8	F2
STOP#	D15	D9	G1
DEVSEL#	BHE#	B5	E4

Core Logic Module (Continued)

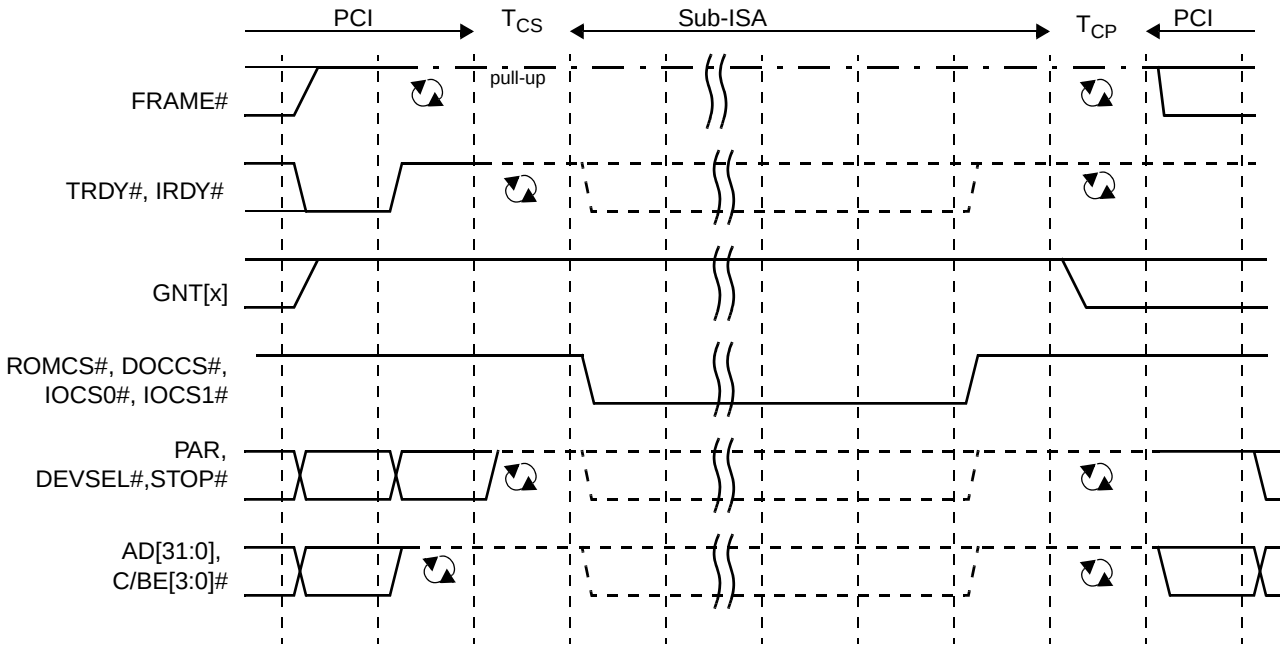


Figure 5-6. PCI Change to Sub-ISA and Back

5.2.6 AT Compatibility Logic

The Core Logic module integrates:

- Two 8237-equivalent DMA controllers with full 32-bit addressing
- Two 8259A-equivalent interrupt controllers providing 13 individually programmable external interrupts
- An 8254-equivalent timer for refresh, timer, and speaker logic
- NMI control and generation for PCI system errors and all parity errors
- Support for standard AT keyboard controllers
- Positive decode for the AT I/O register space
- Reset control

5.2.6.1 DMA Controller

The Core Logic module supports industry standard DMA architecture using two 8237-compatible DMA controllers in cascaded configuration. The DMA functions supported by the Core Logic module include:

- Standard seven-channel DMA support
- 32-bit address range support via high page registers
- IOCHRDY extended cycles for compatible timing transfers
- Internal Sub-ISA bus master device support using cascade mode
- NMI control and generation for PCI system errors and all parity errors

Note: DMA interface signals are not available externally.

DMA Controllers

The Core Logic module supports seven DMA channels using two standard 8237-equivalent controllers. DMA Controller 1 contains Channels 0 through 3 and supports 8-bit I/O adapters. These channels are used to transfer data between 8-bit peripherals and PCI memory or 8/16-bit ISA memory. Using the high and low page address registers, a full 32-bit PCI address is output for each channel so they can all transfer data throughout the entire 4 GB system address space. Each channel can transfer data in 64 KB pages.

DMA Controller 2 contains Channels 4 through 7. Channel 4 is used to cascade DMA Controller 1, so it is not available externally. Channels 5 through 7 support 16-bit I/O adapters to transfer data between 16-bit I/O adapters and 16-bit system memory. Using the high and low page address registers, a full 32-bit PCI address is output for each channel so they can all transfer data throughout the entire 4 GB system address space. Each channel can transfer data in 128 KB pages. Channels 5, 6, and 7 transfer 16-bit WORDs on even byte boundaries only.

DMA Transfer Modes

Each DMA channel can be programmed for *single*, *block*, *demand* or *cascade* transfer modes. In the most commonly used mode, *single* transfer mode, one DMA cycle occurs per DRQ and the PCI bus is released after every cycle. This allows the Core Logic module to timeshare the PCI bus with the GX1 module. This is imperative, especially in cases involving large data transfers, because the GX1 module gets locked out for too long.

Core Logic Module (Continued)

In *block* transfer mode, the DMA controller executes all of its transfers consecutively without releasing the PCI bus.

In *demand* transfer mode, DMA transfer cycles continue to occur as long as DRQ is high or terminal count is not reached. In this mode, the DMA controller continues to execute transfer cycles until the I/O device drops DRQ to indicate its inability to continue providing data. For this case, the PCI bus is held by the Core Logic module until a break in the transfers occurs.

In *cascade* mode, the channel is connected to another DMA controller or to an ISA bus master, rather than to an I/O device. In the Core Logic module, one of the 8237 controllers is designated as the master and the other as the slave. The HOLD output of the slave is tied to the DRQ0 input of the master (Channel 4), and the master's DACK0# output is tied to the slave's HLDA input.

In each of these modes, the DMA controller can be programmed for *read*, *write,* or *verify* transfers.

Both DMA controllers are reset at power-on reset (POR) to *fixed* priority. Since master Channel 0 is actually connected to the slave DMA controller, the slave's four DMA channels have the highest priority, with Channel 0 as highest and Channel 3 as the lowest. Immediately following slave Channel 3, master Channel 1 (Channel 5) is the next highest, followed by Channels 6 and 7.

DMA Controller Registers

The DMA controller can be programmed with standard I/O cycles to the standard register space for DMA. The I/O addresses for the DMA controller registers are listed Table 5-43 on page 302.

When writing to a channel's address or WORD Count register, the data is written into both the base register and the current register simultaneously. When reading a channel address or WORD Count register, only the current address or WORD Count can be read. The base address and base WORD Count are not accessible for reading.

DMA Transfer Types

Each of the seven DMA channels may be programmed to perform one of three types of transfers: *read, write, or verify.* The transfer type selected defines the method used to transfer a byte or WORD during one DMA bus cycle.

For *read* transfer types, the Core Logic module reads data from memory and write it to the I/O device associated with the DMA channel.

For *write* transfer types, the Core Logic module reads data from the I/O device associated with the DMA channel and write to the memory.

The *verify* transfer type causes the Core Logic module to execute DMA transfer bus cycles, including generation of memory addresses, but neither the READ nor WRITE command lines are activated. This transfer type was used by DMA Channel 0 to implement DRAM refresh in the original IBM PC and XT.

DMA Priority

The DMA controller may be programmed for two types of priority schemes: *fixed* and *rotate* (I/O Ports 008h[4] and 0D0h[4] - see Table 5-43 on page 302).

In *fixed* priority, the channels are fixed in priority order based on the descending values of their numbers. Thus, Channel 0 has the highest priority. In *rotate* priority, the last channel to get service becomes the lowest-priority channel with the priority of the others rotating accordingly. This prevents a channel from dominating the system.

The address and WORD Count registers for each channel are 16-bit registers. The value on the data bus is written into the upper byte or lower byte, depending on the state of the internal addressing byte pointer. This pointer can be cleared by the Clear Byte Pointer command. After this command, the first read/write to an address or WORD-count register reads or writes to the low byte of the 16-bit register and the byte pointer points to the high byte. The next read/write to an address or WORD-count register reads or writes to the high byte of the 16-bit register and the byte pointer points back to the low byte.

When programming the 16-bit channels (Channels 5, 6, and 7), the address which is written to the base address register must be the real address divided by two. Also, the base WORD Count for the 16-bit channels is the number of 16-bit WORDs to be transferred, not the number of bytes as is the case for the 8-bit channels.

The DMA controller allows the user to program the active level (low or high) of the DRQ and DACK# signals. Since the two controllers are cascaded together internally on the chip, these signals should always be programmed with the DRQ signal active high and the DACK# signal active low.

DMA Shadow Registers

The Core Logic module contains a shadow register located at F0 Index B8h (Table 5-29 on page 199) for reading the configuration of the DMA controllers. This read only register can sequence to read through all of the DMA registers.

DMA Addressing Capability

DMA transfers occur over the entire 32-bit address range of the PCI bus. This is accomplished by using the DMA controller's 16-bit memory address registers in conjunction with an 8-bit DMA Low Page register and an 8-bit DMA High Page register. These registers, associated with each channel, provide the 32-bit memory address capability. A write to the Low Page register clears the High Page register, for backward compatibility with the PC/AT standard. The starting address for the DMA transfer must be programmed into the DMA controller registers and the channel's respective Low and High Page registers prior to beginning the DMA transfer.

DMA Page Registers and Extended Addressing

The DMA Page registers provide the upper address bits during DMA cycles. DMA addresses do not increment or decrement across page boundaries. Page boundaries for the 8-bit channels (Channels 0 through 3) are every 64 KB and page boundaries for the 16-bit channels (Channels 5, 6, and 7) are every 128 KB.

Before any DMA operations are performed, the Page registers must be written at the I/O Port addresses in the DMA controller registers to select the correct page for each DMA channel. The other address locations between 080h and 08Fh and 480h and 48Fh are not used by the DMA channels, but can be read or written by a PCI bus master. These registers are reset to zero at POR. A write to the Low Page register clears the High Page register, for backward compatibility with the PC/AT standard.

For most DMA transfers, the High Page register is set to zeros and is driven onto PCI address bits AD[31:24] during DMA cycles. This mode is backward compatible with the PC/AT standard. For DMA extended transfers, the High Page register is programmed and the values are driven onto the PCI addresses AD[31:24] during DMA cycles to allow access to the full 4 GB PCI address space.

DMA Address Generation

The DMA addresses are formed such that there is an upper address, a middle address, and a lower address portion.

The upper address portion, which selects a specific page, is generated by the Page registers. The Page registers for each channel must be set up by the system before a DMA operation. The DMA Page register values are driven on PCI address bits AD[31:16] for 8-bit channels and AD[31:17] for 16-bit channels.

The middle address portion, which selects a block within the page, is generated by the DMA controller at the beginning of a DMA operation and any time the DMA address increments or decrements through a block boundary. Block sizes are 256 bytes for 8-bit channels (Channels 0 through 3) and 512 bytes for 16-bit channels (Channels 5, 6, and 7). The middle address bits are is driven on PCI address bits AD[15:8] for 8-bit channels and AD[16:9] for 16-bit channels.

The lower address portion is generated directly by the DMA controller during DMA operations. The lower address bits are output on PCI address bits AD[7:0] for 8-bit channels and AD[8:1] for 16-bit channels.

BHE# is configured as an output during all DMA operations. It is driven as the inversion of AD0 during 8-bit DMA cycles and forced low for all 16-bit DMA cycles.

5.2.6.2 Programmable Interval Timer

The Core Logic module contains an 8254-equivalent Programmable Interval Timer (PIT) configured as shown in Figure 5-7. The PIT has three timers/counters, each with an input frequency of 1.19318 MHz (OSC divided by 12), and individually programmable to different modes.

The gates of Counter 0 and 1 are usually enabled, however, they can be controlled via F0 Index 50h. The gate of Counter 2 is connected to I/O Port 061h[0]. The output of Counter 0 is connected internally to IRQ0. This timer is typically configured in Mode 3 (square wave output), and used to generate IRQ0 at a periodic rate to be used as a system timer function. The output of Counter 1 is connected to I/O Port 061h[4]. The reset state of I/O Port 061h[4] is 0 and every falling edge of Counter 1 output causes I/O Port 061h[4] to flip states. The output of Counter 2 is brought out to the PC_BEEP output. This output is gated with I/O Port 061h[1].

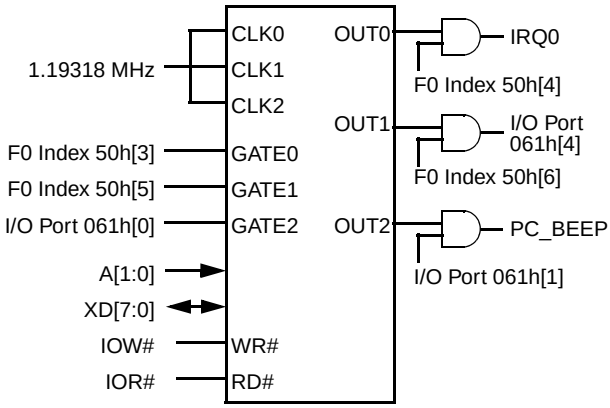


Figure 5-7. PIT Timer

PIT Shadow Register

The PIT registers are shadowed to allow for 0V Suspend to save/restore the PIT state by reading the PIT's counter and *write only* registers. The read sequence for the shadow register is listed in F0 Index BAh (see Table 5-29 on page 199).

Core Logic Module (Continued)

5.2.6.3 Programmable Interrupt Controller

The Core Logic module contains two 8259A-equivalent programmable interrupt controllers, with eight interrupt request lines each, for a total of 16 interrupts. The two controllers are cascaded internally, and two of the interrupt request inputs are connected to the internal circuitry. This allows a total of 13 externally available interrupt requests. See Figure 5-9.

Each Core Logic IRQ signal can be individually selected to as edge- or level-sensitive. The four PCI interrupt signals may be routed internally to any PIC IRQ.

.

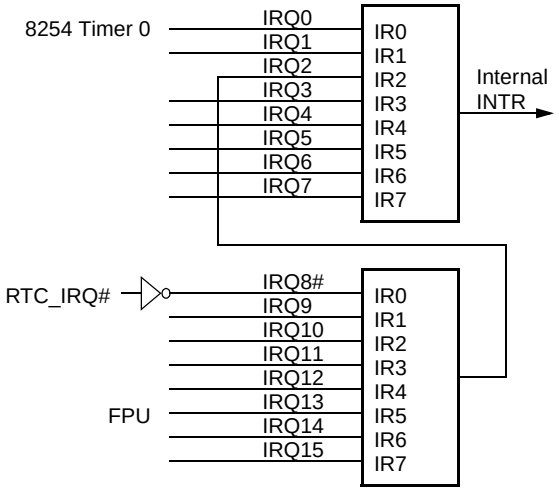


Figure 5-8. PIC Interrupt Controllers

Three interrupts are available externally depending upon selected ball multiplexing:

1) IRQ15 (muxed with GPIO11+RI2#),

2) IRQ14 (muxed with TFTD1), and

3) IRQ9 (muxed with IDE_DATA6)

More of the IRQs are available through the use of SERIRQ (muxed with GPIO39) function. See Table 5-4.

Table 5-4. PIC Interrupt Mapping

Master IRQ	Mapping
IRQ0	Connected to the OUT0 (system timer) of the internal 8254 PIT.
IRQ2	Connected to the slave's INTR for a cascaded configuration.
IRQ8#	Connected to internal RTC.
IRQ13	Connected to the FPU interface of the GX1 module.
IRQ15	Interrupts available to other functions
IRQ14	
IRQ12	
IRQ11	
IRQ10	
IRQ9	
IRQ7	
IRQ6	
IRQ5	
IRQ4	
IRQ3	
IRQ1	

The Core Logic module allows PCI interrupt signals INTA#, INTB#, INTC# (muxed with GPIO19+IOCHRDY) and INTD# (muxed with IDE_DATA7) to be routed internally to any IRQ signal. The routing can be modified through Core Logic module's configuration registers. If this is done, the IRQ input must be configured to be level- rather than edge-sensitive. IRQ inputs may be individually programmed to be active low, level-sensitive with the Interrupt Sensitivity configuration registers at I/O address space 4D0h and 4D1h. PCI interrupt configuration is discussed in further detail in "PCI Compatible Interrupts" on page 165.

Core Logic Module (Continued)

PIC Interrupt Sequence

A typical AT-compatible interrupt sequence is as follows. Any unmasked interrupt generates the internal INTR signal to the CPU. The interrupt controller then responds to the interrupt acknowledge (INTA) cycles from the CPU. On the first INTA cycle the cascading priority is resolved to determine which of the two 8259A controllers output the interrupt vector onto the data bus. On the second INTA cycle the appropriate 8259A controller drives the data bus with the correct interrupt vector for the highest priority interrupt.

By default, the Core Logic module responds to PCI INTA cycles because the system interrupt controller is located within the Core Logic module. This may be disabled with F0 Index 40h[0]. When the Core Logic module responds to a PCI INTA cycle, it holds the PCI bus and internally generates the two INTA cycles to obtain the correct interrupt vector. It then asserts TRDY# and returns the interrupt vector.

PIC I/O Registers

Each PIC contains registers located in the standard I/O address locations, as shown in Table 5-46 "Programmable Interrupt Controller Registers" on page 310.

An initialization sequence must be followed to program the interrupt controllers. The sequence is started by writing Initialization Command Word 1 (ICW1). After ICW1 has been written, the controller expects the next writes to follow in the sequence ICW2, ICW3, and ICW4 if it is needed. The Operation Control Words (OCW) can be written after initialization. The PIC must be programmed before operation begins.

Since the controllers are operating in cascade mode, ICW3 of the master controller should be programmed with a value indicating that the IRQ2 input of the master interrupt controller is connected to the slave interrupt controller rather than an I/O device as part of the system initialization code. In addition, ICW3 of the slave interrupt controller should be programmed with the value 02h (slave ID) and corresponds to the input on the master controller.

PIC Shadow Register

The PIC registers are shadowed to allow for 0V Suspend to save/restore the PIC state by reading the PICs *write only* registers. A write to this register resets the read sequence to the first register. The read sequence for the shadow register is listed in F0 Index B9h.

PCI Compatible Interrupts

The Core Logic module allows the PCI interrupt signals INTA#, INTB#, INTC#, and INTD# (also known in industry terms as PIRQx#) to be mapped internally to any IRQ signal with the PCI Interrupt Steering registers 1 and 2, F0 Index 5Ch and 5Dh.

PCI interrupts are low-level sensitive, whereas PC/AT interrupts are positive-edge sensitive; therefore, the PCI interrupts are inverted before being connected to the 8259A.

Although the controllers default to the PC/AT-compatible mode (positive-edge sensitive), each IRQ may be individually programmed to be edge or level sensitive using the Interrupt Edge/Level Sensitivity registers in I/O Port 4D0h and 4D1h. However, if the controllers are programmed to be level-sensitive via ICW1, all interrupts must be level-sensitive. Figure 5-9 shows the PCI interrupt mapping for the master/slave 8259A interrupt controller.

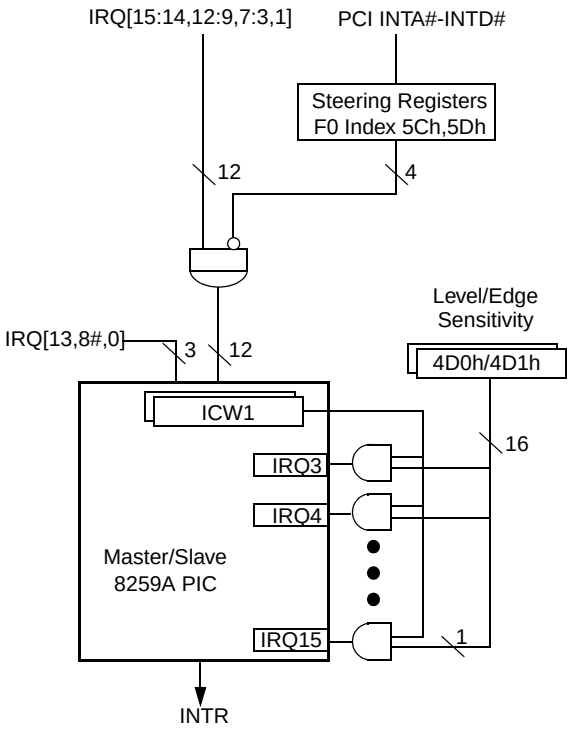


Figure 5-9. PCI and IRQ Interrupt Mapping

5.2.7 I/O Ports 092h and 061h System Control

The Core Logic module supports control functions of I/O Ports 092h (Port A) and 061h (Port B) for PS/2 compatibility. I/O Port 092h allows a fast assertion of the A20M# or CPU_RST. (CPU_RST is an internal signal that resets the CPU. It is asserted for 100 µs after the negation of POR#.) I/O Port 061h controls NMI generation and reports system status.The Core Logic module generates an SMI for every internal change of the A20M# state and the SMI handler sets the A20M# state inside the GX1 module. This method is used for both the Port 092h (PS/2) and Port 061h (keyboard) methods of controlling A20M#.

5.2.7.1 I/O Port 092h System Control

I/O Port 092h allows for a fast keyboard assertion of an A20# SMI and a fast keyboard CPU reset. Decoding for this register may be disabled via F0 Index 52h[3].

The assertion of a fast keyboard A20# SMI is controlled by either I/O Port 092h or by monitoring for the keyboard command sequence (see Section 5.2.8.1 "Fast Keyboard Gate Address 20 and CPU Reset" on page 166). If bit 1 of I/O Port 092h is cleared, the Core Logic module internally asserts an A20M#, which in turn causes an SMI to the GX1 module. If bit 1 is set, A20M# is internally deasserted, again causing an SMI.

Core Logic Module (Continued)

The assertion of a fast keyboard reset (WM_RST SMI) is controlled by bit 0 in I/O Port 092h or by monitoring for the keyboard command sequence (write data = FEh to I/O port 64h). If bit 0 is changed from 0 to 1, the Core Logic module generates a reset to the GX1 module by generating a WM_RST SMI. When the WM_RST SMI occurs, the BIOS jumps to the Warm Reset vector. Note that Warm Reset is not a pin, it is under SMI control.

5.2.7.2 I/O Port 061h System Control

Through I/O Port 061h, the speaker output can be enabled, the status of IOCHK# and SERR# can be read, and the state of the speaker data (Timer2 output) and refresh toggle (Timer1 output) can be read back. Note that NMI is under SMI control. Even though the hardware is present, the IOCHK# ball does not exist. Therefore, an NMI from IOCHK# can not happen.

5.2.7.3 SMI Generation for NMI

Figure 5-10 shows how the Core Logic module can generate an SMI for an NMI. Note that NMI is not a pin.

5.2.8 Keyboard Support

The Core Logic module can actively decode the keyboard controller I/O Ports 060h, 062h, 064h and 066h, and generate an LPC bus cycle. Keyboard positive decoding can be disabled if F0 Index 5Ah[1] is cleared (i.e., subtractive decoding enabled).

5.2.8.1 Fast Keyboard Gate Address 20 and CPU Reset

The Core Logic module monitors the keyboard I/O Ports 064h and 060h for the fast keyboard A20M# and CPU reset control sequences. If a write to I/O Port 060h[1] = 1 after a write takes place to I/O Port 064h with data of D1h, then the Core Logic module asserts the A20M# signal. A20M# remains asserted until cleared by any one of the following:

- A write to bit 1 of I/O Port 092h.
- A CPU reset of some kind.
- A write to I/O Port 060h[1] = 0 following a write to I/O Port 064h with data of D1h.

The fast keyboard A20M# and CPU reset can be disabled through F0 Index 52h[7]. By default, bit 7 is set, and the fast keyboard A20M# and CPU reset monitor logic is active. If bit 7 is clear, the Core Logic module forwards the commands to the keyboard controller.

By default, the Core Logic module forces the de-assertion of A20M# during a warm reset. This action may be disabled if F0 Index 52h[4] is cleared.

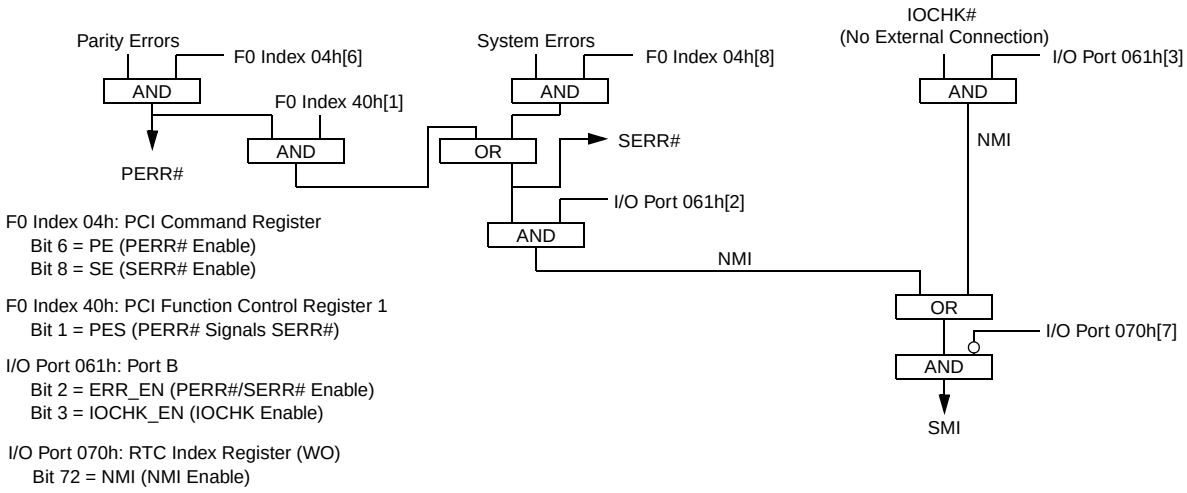


Figure 5-10. SMI Generation for NMI

Core Logic Module (Continued)

5.2.9 Power Management Logic

The Core Logic module integrates advanced power management features including idle timers for common system peripherals, address trap registers for programmable address ranges for I/O or memory accesses, four programmable general purpose external inputs, clock throttling with automatic speedup for the GX1 clock, software GX1 stop clock, 0V Suspend/Resume with peripheral shadow registers, and a dedicated serial bus to/from the GX1 module providing power management status.

The Core Logic module is ACPI (Advanced Configuration Power Interface) compliant. An ACPI-compliant system is one whose underlying BIOS, device drivers, chipset and peripherals conform to revision 1.0 of the ACPI specification. The Core Logic also supports Advanced Power Management (APM).

The SC2200 provides the following support of ACPI states:

- CPU States: C0, C1, and C3.
- Sleep States:
 — SL1/SL2 - ACPI S1 equivalent.
 — SL3 - ACPI S3 equivalent.
 — SL4 - ACPI S4 equivalent.
 — SL5 - ACPI S5 equivalent.
- General Purpose Events: Fully programmable GPE0 Event Block registers.
- Wakeup Events: Supported through GPWIO[2:0] which are powered by standby voltage and generate SMIs. See registers at F1BAR1+I/O Offset 0Ah and F1BAR1+I/O Offset 12h. Also see Section 4.6 "System Wakeup Control (SWC)" on page 125 and Table 5-5 "Wakeup Events Capability" on page 168.

SC2200 device power management is highly tuned for low power systems. It allows the system designer to implement a wide range of power saving modes using a wide range of capabilities and configuration options.

SC2200 controls the following functions directly:

- The system clocks.
- Core processor power states.
- Wakeup/resume event detection, including general purpose events.
- Power supply and power planes.

It also supports systems with an external micro controller that is used as a power management controller.

5.2.9.1 CPU States

The SC2200 supports three CPU states: C0, C1 and C3 (the Core Logic C2 CPU state is not supported). These states are fully compliant with the ACPI specification, revision 1.0. These states occur in the Working state only (S0/G0). They have no meaning when the system transitions into a Sleep state. For details on the various Sleep states, see Section 5.2.9.2 "Sleep States" on page 167.

C0 Power State - On

In this state the GX1 module executes code. This state has two sub-states: Full Speed or Throttling; selected via the THT_EN bit (F1BAR1+I/O Offset 00h[4]).

C1 Power State - Active Idle

The SC2200 enters the C1 state, when the Halt Instruction (HLT) is executed. It exits this state back to the C0 state upon an NMI, an unmasked interrupt, or an SMI. The Halt instruction stops program execution and generates a special Halt bus cycle. (See “Usage Hints” on page 170.)

Bus masters are supported in the C1 state and the SC2200 will temporarily exit C1 to perform a bus master transaction.

C2 Power State

The SC2200 does not support the C2 power state. All relevant registers and bit fields in the Core Logic are reserved.

C3 Power State

The SC2200 enters the C3 state, when the P_LVL3 register (F1BAR1+I/O Offset 05h) is read. It exits this state back to the C0 state (Full Speed or Throttling, depending on the THT_EN bit) upon:

- An NMI, an unmasked interrupt, or an SMI.
- A bus master request, if enabled via the BM_RLD bit (F1BAR1+I/O Offset 0Ch[1]).

In this state, the GX1 module is in Suspend Refresh mode (for details, see the Power Management section of the *GX1 Processor Series Datasheet*, and Section 5.2.9.5 "Usage Hints" on page 170).

PCI arbitration should be disabled prior entering the C3 state via the ARB_DIS bit in the PM2_CNT register (F1BAR1+I/O Offset 20h[0]) because a PCI arbitration event could start after P_LVL3 has been read. After wakeup ARB_DIS needs to be cleared.

5.2.9.2 Sleep States

The SC2200 supports four Sleep states (SL1-SL3) and the Soft Off state (G2/S5). These states are fully compliant with the ACPI specification, revision 1.0.

When the SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]) is set to 1, the SC2200 enters an SLx state according to the SLP_TYPx field (F1BAR1+I/O Offset 0Ch[12:10]). It exits the Sleep state back to the S0 state (C0 state - Full Speed or Throttling, depending on the THT_EN bit) upon an enabled power management event. Table 5-5 on page 168 lists wakeup events from the various Sleep states.

Core Logic Module (Continued)

SL1 Sleep State (ACPI S1)

In this state the core processor is in 3V Suspend mode (all its clocks are stopped, including the memory controller and the display controller). The SDRAM is placed in self-refresh mode. All other SC2200 system clocks and PLLs are running. All devices are powered up (PWRCNT[2:1] and ONCTL# are all asserted). See Section 5.2.9.5 "Usage Hints" on page 170.

No reset is performed, when exiting this state. The SC2200 keeps all context in this state. This state corresponds to ACPI Sleep state S1.

SL2 Sleep State (ACPI S1)

In this state, all of the SC2200 clocks are stopped including the PLLs. Selected clocks from the PLLs can be kept running under program control (F0 Index 60h). An exception to this is the CLK32 output signal which keeps toggling and the 32 KHz oscillator itself. The SDRAM is placed in self-refresh mode. The PWRCNT1 pin is de-asserted. The SC2200 itself is powered up. The system designer can decide which other system devices to power off with the PWRCNT1 pin.

No reset is performed, when exiting this state. The SC2200 keeps all context in this state. This state corresponds to ACPI sleep state S1, with lower power and longer wake time than in SL1.

SL3 Sleep State (ACPI S3)

In this state, the SDRAM is placed in self-refresh mode, and PWRCNT[2:1] are de-asserted. PWRCNT[2:1] should be used to power off most of the system (except for the SDRAM). If the Save-to-RAM feature is used, external circuitry in the SDRAM interface is required to guarantee data integrity. All SC2200 signals powered by V_{SB}, V_{SBL} or V_{BAT} are still functional to allow wakeup and to keep the RTC.

The power-up sequence is performed, when exiting this state. This state corresponds to ACPI Sleep state S3.

SL4 and SL5 Sleep States (ACPI S4 and S5)

The SL4 and SL5 states are similar from the hardware perspective. In these states, the SC2200 de-asserts PWRCNT[2:1] and ONCTL#. PWRCNT[2:1] and ONCTL# should be used to power off the system. All signals powered by V_{SB}, V_{SBL} or V_{BAT} are still functional to allow wakeup and to keep the RTC.

While in this state, LED# can be toggled to give visual notification of this state. ACPI Function Control register (F1BAR1+I/O Offset 07h[7:6]) is used to control LED#.

The power-up sequence is performed when exiting this state. This state corresponds to ACPI Sleep states S4 and S5.

Table 5-5. Wakeup Events Capability

Event	S0/C1	S0/C3	SL1	SL2	SL3	SL4, SL5
Enabled Interrupts	Yes	Yes	Yes	-	-	-
SMI according to Table 5-8	Yes	Yes	Yes	-	-	-
SCI according to Table 5-8	Yes	Yes	Yes	-	-	-
GPIO[47:32], GPIO[15:0]	Yes	Yes	Yes	-	-	-
Power Button	Yes	Yes	Yes	Yes	Yes	Yes
Power Button Override	Yes	Yes	Yes	Yes	Yes	Yes
Bus Master Request	Yes[1]	Yes	Yes	-	-	-
Thermal Monitoring	Yes	Yes	Yes	Yes	Yes	Yes
USB	Yes	Yes	Yes	Yes	-	-
SDATA_IN2 (AC97)	Yes	Yes	Yes	Yes	-	-
IRRX1 (Infrared)	Yes	Yes	Yes	Yes	-	-
GPWIO[2:0]	Yes	Yes	Yes	Yes	Yes	Yes
RI2# (UART2)	Yes	Yes	Yes	Yes	-	-
RTC	Yes	Yes	Yes	Yes	Yes	Yes

1. Temporarily exits state.

Core Logic Module (Continued)

5.2.9.3 Power Planes Control

The SC2200 supports up to three power planes. Three signals are used to control these power planes. Table 5-6 describes the signals and when each is asserted.

Table 5-6. Power Planes Control Signals vs. Sleep States

Signal	S0	SL1	SL2	SL3	SL4 and SL5
PWRCNT1	1	1	0	0	0
PWRCNT2	1	1	1	0	0
ONCTL#	0	0	0	0	1

These signals allow control of the power of system devices and the SC2200 itself. Table 5-7 describes the SC2200 power planes with respect to the different Sleep and Global states.

Table 5-7. Power Planes vs. Sleep/Global States

Sleep/ Global State	V_{CORE}, V_{CCCRT}, $V_{I/O}$, V_{PLL}, AV_{CCCRT}	V_{SB}, V_{SBL}	V_{BAT}
S0, SL1 and SL2	On	On	On or Off
SL3, SL4 and SL5	Off	On	On or Off
G3	Off	Off	On
No Power	Off	Off	Off
Illegal	On	Off	On or Off

The SC2200 power planes are controlled externally by the three signals (i.e., the system designer should make sure the system design is such that Table 5-7 is met) for all supported Sleep states.

V_{SB} and V_{BAT} are not controlled by any control signal. V_{SB} exists as long as the AC power is plugged in (for desktop systems) or the main battery is charged (for mobile systems). V_{BAT} exists as long as the RTC battery is charged.

The case in which V_{SB} does not exist is called Mechanical Off (G3).

5.2.9.4 Power Management Events

The SC2200 supports power management events that can manage:

- Transition of the system from a Sleep state to a Work state. This is done by the hardware. These events are defined as wakeup events.
- Enabled wakeup events to set the WAK_STS bit (F1BAR1+I/O Offset 08h[15]) to 1, when transitioning the system back to the working state.
- Generation of an interrupt. This invokes the relevant software driver. The interrupt can either be an SMI or SCI (selected by the SCI_EN bit, F1BAR1+I/O Offset 0Ch[0]). These events are defined as interrupt events.

Table 5-8 lists the power management events that can generate an SCI or SMI.

Table 5-8. Power Management Events

Event	SCI	SMI
Power Button	Yes	Yes
Power Button Override	Yes	-
Bus Master Request	Yes	-
Thermal Monitoring	Yes	Yes
USB	Yes	Yes
RTC	Yes	Yes
ACPI Timer	Yes	Yes
GPIO	Yes	Yes
SDATA_IN2 (AC97)	Yes	Yes
IRRX1	Yes	Yes
RI2#	Yes	Yes
GPWIO	Yes	Yes
Internal SMI signal	Yes	-

Power Button

The power button (PWRBTN#) input provides two events: a wake request, and a sleep request. For both these events, the PWRBTN# signal is debounced (i.e., the signal state is transferred only after 14 to 16 ms without transitions, to ensure that the signal is no longer bouncing).

ACPI is non-functional when the power-up sequence does not include using the power button. If ACPI functionality is desired, the power button must be toggled. This can be done externally or internally. GPIO63 is internally connected to PWRBTN#. To toggle the power button with software, GPIO63 must be programmed as an output using the normal GPIO programming protocol (see Section 5.4.1.1 "GPIO Support Registers" on page 231). GPIO63 must be pulsed low for at least 16 ms and not more than 4 sec. Asserting POR# has no effect on ACPI. If POR# is asserted and ACPI was active prior to POR#, then ACPI

will remain active after POR#. Therefore, BIOS must ensure that ACPI is inactive before GPIO63 is pulsed low.

Power Button Wake Event - Detection of a high-to-low transition on the debounced PWRBTN# input signal when in SL1 to SL5 Sleep states. The system is considered in the Sleep state, only after it actually transitioned into the state and not only according to the SLP_TYP field.

In reaction to this event, the PWRBTN_STS bit (F1BAR1+I/O Offset 08h[8]) is set to 1 and a wakeup event or an interrupt is generated (note that this is regardless of the PWRBTN_EN bit, F1BAR1+I/O Offset 0Ah[8]).

Power Button Sleep Event - Detection of a high-to-low transition on the debounced PWRBTN# input signal, when in the Working state (S0).

In reaction to this event, the PWRBTN_STS bit is set to 1.

- When both the PWRBTN_STS bit and the PWRBTN_EN bit are set to 1, an SCI interrupt is generated.
- When SCI_EN bit is 0, ONCTL# and PWRCNT[2:1] are de-asserted immediately regardless of the PWRBTN_EN bit.

Power Button Override

When PWRBTN# is 0 for more than four seconds, ONCTL# and PWRCNT[2:1] are de-asserted (i.e., the system transitions to the SL5 state, “Soft Off”). This power management event is called the power button override event.

In reaction to this event, the PWRBTN_STS bit is cleared to 0 and the PWRBTNOR_STS bit (F1BAR1+I/O Offset 08h[11]) is set to 1.

Thermal Monitoring

The thermal monitoring event (THRM#) enables control of ACPI-OS Control.

When the THRM# signal transitions from high-to-low, the THRM_STS bit (F1BAR1+I/O Offset 10h[5]) is set to 1. If the THRM_EN bit (F1BAR1+I/O Offset 12h[5]) is also set to 1, an interrupt is generated.

SDATA_IN2, IRRX1, RI2#

Section 4.4.1 "SIO Control and Configuration Registers" on page 106 for control and operation.

5.2.9.5 Usage Hints

- During initialization, the BIOS should:
 - — Clear the SUSP_HLT bit in CCR2 (GX1 module, Index C2h[3]) to 0. This is needed for compliance with C0 definition of ACPI, when the Halt Instruction (HLT) is executed.
 - — Disable the SUSP_3V option in C3 power state (F0 Index 60h[2]).
 - — Disable the SUSP_3V option in SL1 sleep state (F0 Index 60h[1]).
- SMM code should clear the CLK_STP bit in the PM Clock Stop Control register (GX_BASE+Memory Offset 8500h[0]) to 0 when entering C3 state.
- SMM code should correctly set the CLK_STP bit in the PM Clock Stop Control register (GX_BASE+Memory Offset 8500h[0]) when entering the SL1, SL2, and SL3 states.

5.2.10 Power Management Programming

The power management resources provided by a combined GX1 module and Core Logic module based system supports a high efficiency power management implementation. The following explanations pertain to a full-featured “notebook” power management system. The extent to which these resources are employed depends on the application and on the discretion of the system designer.

Power management resources can be grouped according to the function they enable or support. The major functions are as follows:

- APM Support
- CPU Power Management
 - — Suspend Modulation
 - — 3V Suspend
 - — Save-to-Disk
- Peripheral Power Management
 - — Device Idle Timers and Traps
 - — General Purpose Timers
 - — ACPI Timer Register
 - — Power Management SMI Status Reporting Registers

Included in the following subsections are details regarding the registers used for configuring power management features. The majority of these registers are directly accessed through the PCI configuration register space designated as Function 0 (F0). However, included in the discussions are references to F1BARx+I/O Offset xxh. This refers to registers accessed through base address registers in Function 1 (F1) at Index 10h (F1BAR0) and Index 40h (F1BAR1).

5.2.10.1 APM Support

Many notebook computers rely solely on an Advanced Power Management (APM) driver for enabling the operating system to power-manage the CPU. APM provides several services which enhance the system power management; but in its current form, APM is imperfect for the following reasons:

- APM is an OS-specific driver, and may not be available for some operating systems.
- Application support is inconsistent. Some applications in foreground may prevent Idle calls.
- APM does not help with Suspend determination or peripheral power management.

The Core Logic module provides two entry points for APM support:

- Software CPU Suspend control via the CPU Suspend Command register (F0 Index AEh).
- Software SMI entry via the Software SMI register (F0 Index D0h). This allows the APM BIOS to be part of the SMI handler.

Core Logic Module (Continued)

5.2.10.2 CPU Power Management

The three greatest power consumers in a system are the display, the hard drive, and the CPU. The power management of the first two is relatively straightforward and is discussed in Section 5.2.10.3 "Peripheral Power Management" on page 172.

APM, if available, is used primarily by CPU power management since the operating system is most capable of reporting the Idle condition. Additional resources provided by the Core Logic module supplement APM by monitoring external activity and power managing the CPU based on the system demands. The two processes for power managing the CPU are Suspend Modulation and 3V Suspend.

Suspend Modulation

Suspend Modulation works by asserting and de-asserting the internal SUSP# signal to the GX1 module for configurable durations. When SUSP# is asserted to the GX1 module, it enters an Idle state during which time the power consumption is significantly reduced. Even though the PCI clock is still running, the GX1 module stops the clocks to its core when SUSP# is asserted. By modulating SUSP# a reduced frequency of operation is achieved.

The Suspend Modulation feature works by assuming that the GX1 module is Idle unless external activity indicates otherwise. This approach effectively slows down the GX1 module until external activity indicates a need to run at full speed, thereby reducing power consumption. This approach is the opposite of that taken by most power management schemes in the industry, which run the system at full speed until a period of inactivity is detected, and then slows down. Suspend Modulation, the more aggressive approach, yields lower power consumption.

Suspend Modulation serves as the primary CPU power management mechanism when APM is not present. It also acts as a backup for situations where APM does not correctly detect an Idle condition in the system.

To provide high-speed performance when needed, SUSP# modulation is temporarily disabled any time system activity is detected. When this happens, the GX1 module is “instantly” converted to full speed for a programmed duration. System activities in the Core Logic module are asserted as: any unmasked IRQ, accessing Port 061h, any asserted SMI, and/or accessing the Video Processor module interface.

The graphics controller is integrated in the GX1 module. Therefore, the indication of video activity is sent to the Core Logic module via the serial link (see Section 5.2.2 "PSERIAL Interface" on page 152 for more information on serial link) and is automatically decoded. Video activity is defined as any access to the VGA register space, the VGA frame buffer, the graphics accelerator control registers and the configured graphics frame buffer.

The automatic speedup events (video and IRQ) for Suspend Modulation should be used together with software-controlled speedup registers for major I/O events such as any access to the FDC, HDD, or parallel/serial ports, since these are indications of major system activities. When major I/O events occur, Suspend Modulation should be temporarily disabled using the procedures described in the Power Management registers in the following subsections.

If a bus master (UltraDMA/33, Audio, USB) request occurs, the GX1 module automatically de-asserts SUSPA# and grants the bus to the requesting bus master. When the bus master de-asserts REQ#, SUSPA# reasserts. This does not directly affect the Suspend Modulation programming.

Configuring Suspend Modulation: Control of the Suspend Modulation feature is accomplished using the Suspend Modulation and Suspend Configuration registers (F0 Index 94h and 96h, respectively).

The Suspend Configuration register contains the global power management enable bit, as well as the enables for the individual activity speedup timers. The global power management bit must be enabled for Suspend Modulation and all other power management resources to function.

Bit 0 of the Suspend Configuration register enables Suspend Modulation. Bit 1 controls how SMI events affect Suspend Modulation. In general this bit should be set to 1, which causes SMIs to disable Suspend Modulation until it is re-enabled by the SMI handler.

The Suspend Modulation register controls two 8-bit counters that represent the number of 32 µs intervals that the internal SUSP# signal is asserted and then de-asserted to the GX1 module. These counters define a ratio which is the effective frequency of operation of the system while Suspend Modulation is enabled.

$$F_{eff} = F_{GX1} \times \frac{\text{Asserted Count}}{\text{Asserted Count + De-asserted Count}}$$

The IRQ and Video Speedup Timer Count registers (F0 Index 8Ch and 8Dh) configure the amount of time which Suspend Modulation is disabled when the respective events occur.

SMI Speedup Disable: If the Suspend Modulation feature is being used for CPU power management, the occurrence of an SMI disables Suspend Modulation so that the system operates at full speed while in SMM. There are two methods used to invoke this via bit 1 of the Suspend Configuration register.

1) If F0 Index 96h[1] = 0: Use the IRQ Speedup Timer (F0 Index 8Ch) to temporarily disable Suspend Modulation when an SMI occurs.
2) If F0 Index 96h[1] = 1: Disable Suspend Modulation when an SMI occurs until a read to the SMI Speedup Disable register (F1BAR0+I/O Offset 08h).

The SMI Speedup Disable register prevents VSA software from entering Suspend Modulation while operating in SMM. The data read from this register can be ignored. If the Suspend Modulation feature is disabled, reading this I/O location has no effect.

Core Logic Module (Continued)

3 Volt Suspend

The Core Logic module supports the stopping of the CPU and system clocks for a 3V Suspend state. If appropriately configured, via the Clock Stop Control register (F0 Index BCh), the Core Logic module asserts internal SUSP_3V after it has gone through the SUSP#/SUSPA# handshake. SUSP_3V is a state indicator, indicating that the system is in a low-activity state and Suspend Modulation is active. This indicator can be used to put the system into a low-power state (the system clock can be turned off).

Internal SUSP_3V is connected to the enable control of the clock generators, so that the clocks to the CPU and the Core Logic module (and most other system devices) are stopped. The Core Logic module continues to decrement all of its device timers and respond to external SMI interrupts after the input clock has been stopped, as long as the 32 KHz clock continues to oscillate. Any SMI event or unmasked interrupt causes the Core Logic module to de-assert SUSP_3V, restarting the system clocks. As the CPU or other device might include a PLL, the Core Logic module holds SUSP# active for a pre-programmed period of delay (the PLL re-sync delay) that varies from 0 to 15 ms. After this period has expired, the Core Logic module de-asserts SUSP#, stopping Suspend. SMI# is held active for the entire period, so that the CPU reenters SMM when the clocks are restarted.

Save-to-Disk

Save-to-Disk is supported by the Core Logic module. In this state, the power is typically removed from the Core Logic module and from the entire SC2200, causing the state of the legacy peripheral devices to be lost. Shadow registers are provided for devices which allow their state to be saved prior to removing power. This is necessary because the legacy AT peripheral devices used several write only registers. To restore the exact state of these devices on resume, the write only register values are “shadowed” so that the values can be saved by the power management software.

The PC/AT compatible keyboard controller (KBC) and floppy port (FDC) do not exist in the SC2200. However, it is possible that one is attached on the ISA bus or the LPC bus (e.g., in a SuperI/O device). Some of the KBC and FDC registers are shadowed because they cannot be safely read. Additional shadow registers for other functions are described in Table 5-29 "F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support" on page 199.

5.2.10.3 Peripheral Power Management

The Core Logic module provides peripheral power management using a combination of device idle timers, address traps, and general purpose I/O pins. Idle timers are used in conjunction with traps to support powering down peripheral devices.

Device Idle Timers and Traps

Idle timers are used to power manage a peripheral by determining when the peripheral has been inactive for a specified period of time, and removing power from the peripheral at the end of that time period.

Idle timers are provided for the commonly-used peripherals (FDC, IDE, parallel/serial ports, and mouse/keyboard). In addition, there are three user-defined timers that can be configured for either I/O or memory ranges.

The idle timers are 16-bit countdown timers with a one second time base, providing a timeout range of 1 to 65536 seconds (1092 minutes) (18 hours).

When the idle timer count registers are loaded with a non-zero value and enabled, the timers decrement until one of two possibilities happens: a bus cycle occurs at that I/O or memory range, or the timer decrements to zero.

If a bus cycle occurs, the timer is reloaded and begins decrementing again. If the timer decrements to zero, and power management is enabled (F0 Index 80h[0] = 1), the timer generates an SMI.

When an idle timer generates an SMI, the SMI handler manages the peripheral power, disables the timer, and enables the trap. The next time an event occurs, the trap generates an SMI. This time, the SMI handler applies power to the peripheral, resets the timer, and disables the trap.

Relevant registers for controlling Device Idle Timers are: F0 Index 80h, 81h, 82h, 93h, 98h-9Eh, and ACh.

Relevant registers for controlling User Defined Device Idle Timers are: F0 Index 81h, 82h, A0h, A2h, A4h, C0h, C4h, C8h, CCh, CDh, and CEh.

Although not considered as device idle timers, two additional timers are provided by the Core Logic module. The Video Idle Timer used for Suspend-determination and the VGA Timer used for SoftVGA.

The programming bits for these timers are:

- F0 Index 81h[7], Video Access Idle Timer Enable
- F0 Index 82h[7], Video Access Trap Enable
- F0 Index A6h[15:0], Video Timer Count
- F0 Index 83h[3], VGA Timer Enable
- F0 Index 8Bh[6], VGA Timer Base
- F0 Index 8Eh[7:0], VGA Timer Count

Core Logic Module (Continued)

General Purpose Timers

The Core Logic module contains two general purpose idle timers, General Purpose Timer 1 (F0 Index 88h) and General Purpose Timer 2 (F0 Index 8Ah). These two timers are similar to the Device Idle Timers in that they count down to zero unless re-triggered, and generate an SMI when they reach zero. However, these are 8-bit timers instead of 16 bits, they have a programmable timebase, and the events which reload these timers are configurable. These timers are typically used for an indication of system inactivity for Suspend determination.

General Purpose Timer 1 can be re-triggered by activity to any of the configured User Defined Devices, Keyboard and Mouse, Parallel and Serial, Floppy disk, or Hard disk.

General Purpose Timer 2 can be re-triggered by a transition on the GPIO7 signal (if GPIO7 is properly configured).

When a General Purpose Timer is enabled or when an event reloads the timer, the timer is loaded with the configured count value. Upon expiration of the timer an SMI is generated and a status flag is set. Once expired, this counter must be re-initialized by disabling and enabling it.

The timebase for both General Purpose Timers can be configured as either 1 second (default) or 1 millisecond. The registers at F0 Index 89h and 8Bh are the control registers for the General Purpose Timers.

ACPI Timer Register

The ACPI Timer register (F1BAR0+I/O Offset 1Ch or at F1BAR1+I/O Offset 1Ch) provides the ACPI counter. The counter counts at 14.31818/4 MHz (3.579545 MHz). If SMI generation is enabled (F0 Index 83h[5] = 1), an SMI or SCI is generated when bit 23 toggles.

Power Management SMI Status Reporting Registers

The Core Logic module updates status registers to reflect the SMI sources. Power management SMI sources are the device idle timers, address traps, and general purpose I/O pins.

Power management events are reported to the GX1 module through the active low SMI# signal. When an SMI is initiated, the SMI# signal is asserted low and is held low until all SMI sources are cleared. At that time, SMI# is deasserted.

All SMI sources report to the Top Level SMI Status register (F1BAR0+I/O Offset 02h) and the Top Level SMI Status Mirror register (F1BAR0+I/O Offset 00h). The Top SMI Status and Status Mirror registers are the top level of hierarchy for the SMI Handler in determining the source of an SMI. These two registers are identical except that reading the register at F1BAR0+I/O Offset 02h clears the status.

Since all SMI sources report to the Top Level SMI Status register, many of its bits combine a large number of events requiring a second level of SMI status reporting. The second level of SMI status reporting is set up very much like the top level. There are two status reporting registers, one “read only” (mirror) and one “read to clear”. The data returned by reading either offset is the same, the difference between the two being that the SMI can not be cleared by reading the mirror register.

Figure 5-11 on page 174 shows an example SMI tree for checking and clearing the source of General Purpose Timers and the User Defined Trap generated SMI.

Core Logic Module (Continued)

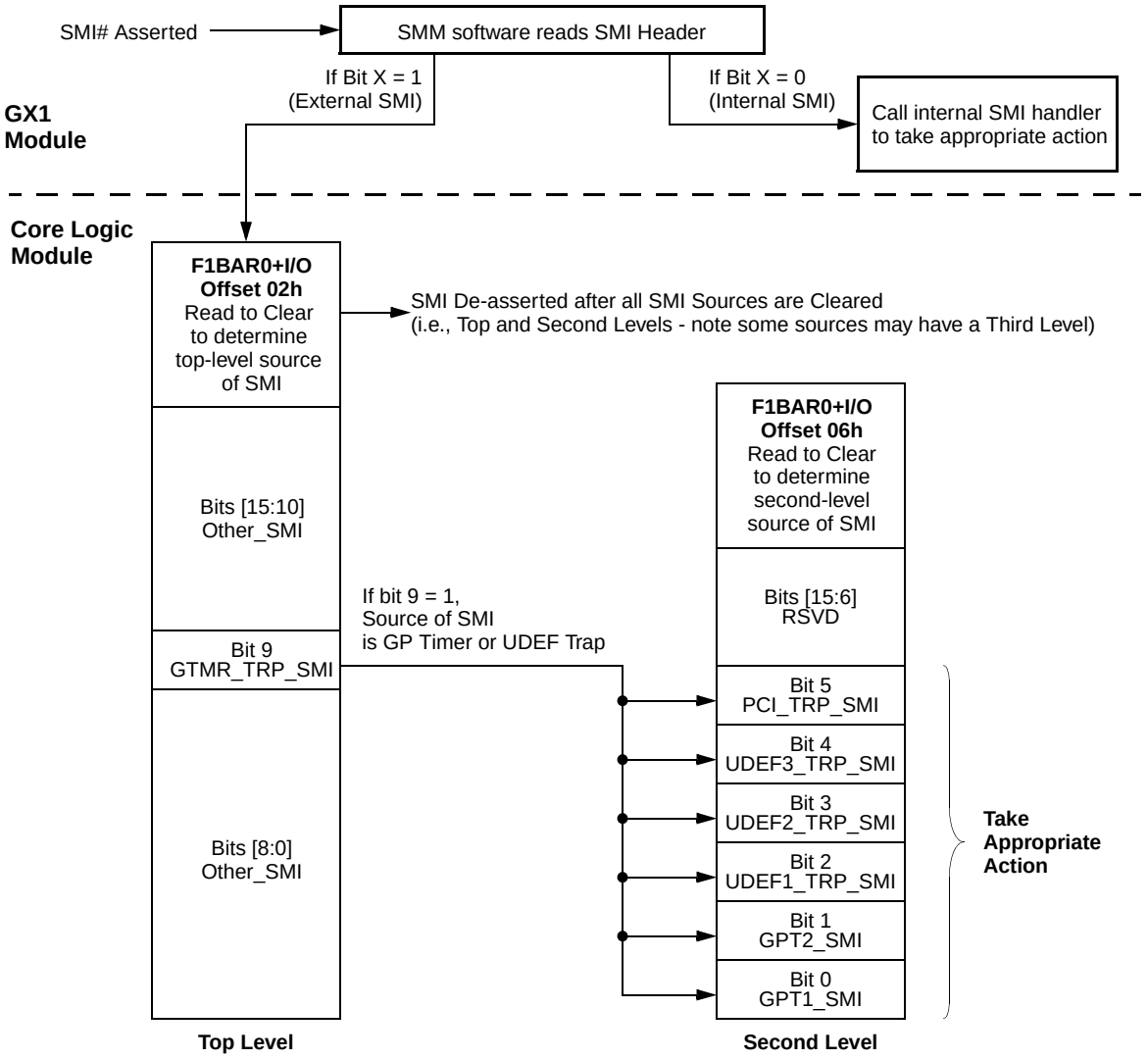


Figure 5-11. General Purpose Timer and UDEF Trap SMI Tree Example

Core Logic Module (Continued)

5.2.10.4 Power Management Programming Summary

Table 5-9 provides a programming register summary for the power management timers, traps, and functions. For complete bit information regarding the registers listed in Table 5-9, refer to Section 5.4.1 "Bridge, GPIO, and LPC Registers - Function 0" on page 199.

Table 5-9. Device Power Management Programming Summary

Device Power Management Resource	Located at F0 Index xxh Unless Otherwise Noted			
	Enable	Configuration	Second Level SMI Status/No Clear	Second Level SMI Status/With Clear
Global Timer Enable	80h[0]	N/A	N/A	N/A
Keyboard / Mouse Idle Timer	81h[3]	93h[1:0]	85h[3]	F5h[3]
Parallel / Serial Idle Timer	81h[2]	93h[1:0]	85h[2]	F5h[2]
Floppy Disk Idle Timer	81h[1]	9Ah[15:0], 93h[7]	85h[1]	F5h[1]
Video Idle Timer[1]	81h[7]	A6h[15:0]	85h[7]	F5h[7]
VGA Timer[2]	83h[3]	8Eh[7:0]	F1BAR0+I/O Offset 00h[6]	F1BAR0+I/O Offset 02h[6]
Primary Hard Disk Idle Timer	81h[0]	98h[15:0], 93h[5]	85h[0]	F5h[0]
Secondary Hard Disk Idle Timer	83h[7]	ACh[15:0], 93h[4]	86h[4]	F6h[4]
User Defined Device 1 Idle Timer	81h[4]	A0h[15:0], C0h[31:0], CCh[7:0]	85h[4]	F5h[4]
User Defined Device 2 Idle Timer	81h[5]	A2h[15:0], C4h[31:0], CDh[7:0]	85h[5]	F5h[5]
User Defined Device 3 Idle Timer	81h[6]	A4h[15:0], C8h[31:0], CEh[7:0]	85h[6]	F5h[6]
Global Trap Enable	80h[2]	N/A	N/A	N/A
Keyboard / Mouse Trap	82h[3]	9Eh[15:0] 93h[1:0]	86h[3]	F6h[3]
Parallel / Serial Trap	82h[2]	9Ch[15:0], 93h[1:0]	86h[2]	F6h[2]
Floppy Disk Trap	82h[1]	93h[7]	86h[1]	F6h[1]
Video Access Trap	82h[7]	N/A	86h[7]	F6h[7]
Primary Hard Disk Trap	82h[0]	93h[5]	86h[0]	F6h[0]
Secondary Hard Disk Trap	83h[6]	93h[4]	86h[5]	F6h[5]
User Defined Device 1 Trap	82h[4]	C0h[31:0], CCh[7:0]	F1BAR0+I/O Offset 04h[2]	F1BAR0+I/O Offset 06h[2]
User Defined Device 2 Trap	82h[5]	C4h[31:0], CDh[7:0]	F1BAR0+I/O Offset 04h[3]	F1BAR0+I/O Offset 06h[3]
User Defined Device 3 Trap	82h[6]	C8h[31:0], CEh[7:0]	F1BAR0+I/O Offset 04h[4]	F1BAR0+I/O Offset 06h[4]
General Purpose Timer 1	83h[0]	88h[7:0], 89h[7:0], 8Bh[4]	F1BAR0+I/O Offset 04h[0]	F1BAR0+I/O Offset 06h[0]
General Purpose Timer 2	83h[1]	8Ah[7:0], 8Bh[5,3,2]	F1BAR0+I/O Offset 04h[1]	F1BAR0+I/O Offset 06h[1]
Suspend Modulation	96h[0]	94h[15:0], 96h[2:0]	N/A	N/A
Video Speedup	80h[4]	8Dh[7:0], A8h[15:0]	N/A	N/A
IRQ Speedup	80h[3]	8Ch[7:0]	N/A	N/A

1. This function is used for Suspend determination.
2. This function is used for SoftVGA.

Core Logic Module (Continued)

5.2.11 GPIO Interface

Up to 64 GPIOs in the in the Core Logic module are provided for system control. For further information, see Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 81 and Table 5-30 "F0BAR0+I/O Offset: GPIO Configuration Registers" on page 231.

Note: Not all GPIOs are available on SC2200 balls. GPIOs [63:42], [31:21], and [5:2] are reserved.

5.2.12 Integrated Audio

The Core Logic module provides hardware support for the Virtual (soft) Audio subsystem as part of the Virtual System Architecture® (VSA™) technology for capture and playback of audio using an external codec. This eliminates much of the hardware traditionally associated with audio functions.

This hardware support includes:

- Six-channel buffered PCI bus mastering interface.
- AC97 version 2.0 compatible interface to the codec. Any codec, which supports an independent input and output sample rate conversion interface, can be used with the Core Logic module.

Additional hardware provides the necessary functionality for VSA. This hardware includes the ability to:

- Generate an SMI to alert software to update required data. An SMI is generated when either audio buffer is half empty or full. If the buffers become completely empty or full, the Empty bit is asserted.
- Generate an SMI on I/O traps.
- Trap accesses for sound card compatibility at either I/O Port 220h-22Fh, 240h-24Fh, 260h-26Fh, or 280h-28Fh.
- Trap accesses for FM compatibility at I/O Port 388h-38Bh.
- Trap accesses for MIDI UART interface at I/O Port 300h-301h or 330h-331h.
- Trap accesses for serial input and output at COM2 (I/O Port 2F8h-2FFh) or COM4 (I/O Port 2E8h-2EFh).
- Support trapping for low (I/O Port 00h-0Fh) and/or high (I/O Port C0h-DFh) DMA accesses.
- Support hardware status register reads in Core Logic module, minimizing SMI overhead.
- Support is provided for software-generated IRQs on IRQ 2, 3, 5, 7, 10, 11, 12, 13, 14, and 15.

The following subsections include details of the registers used for configuring the audio interface. The registers are accessed through F3 Index 10h, the Base Address Register (F3BAR0) in Function 3. F3BAR0 sets the base address for the audio support registers as shown in Table 5-37 "F3: PCI Header Registers for Audio Configuration" on page 269.

5.2.12.1 Data Transport Hardware

The data transport hardware can be broadly divided into two sections: bus mastering and the codec interface.

Audio Bus Masters

The Core Logic module audio hardware includes six PCI bus masters (three for input and three for output) for transferring digitized audio between memory and the external codec. With these bus master engines, the Core Logic module off-loads the CPU and improves system performance.

The programming interface defines a simple scatter/gather mechanism allowing large transfer blocks to be scattered to or gathered from memory. This cuts down on the number of interrupts to and interactions with the CPU.

The six bus masters that directly drive specific slots on the AC97 interface are described in Table 5-10.

Table 5-10. Bus Masters That Drive Specific Slots of the AC97 Interface

Audio Bus Master #	Slots	Description
0	3 and 4	32-Bit output to codec. Left and right channels.
1	3 and 4	32-Bit input from codec. Left and right channels.
2	5	16-Bit output to codec.
3	5	16-Bit input from codec.
4	6 or 11	16-Bit output to codec. Slot in use is determined by F3BAR0+Memory Offset 08h[19].
5	6 or 11	16-Bit input from codec. Slot in use is determined by F3BAR0+Memory Offset 08h[20].

Core Logic Module (Continued)

Physical Region Descriptor Table Address

Before the bus master starts a master transfer it must be programmed with a pointer (PRD Table Address register) to a Physical Region Descriptor Table. This pointer sets the starting memory location of the Physical Region Descriptors (PRDs). The PRDs describe the areas of memory that are used in the data transfer. The descriptor table entries must be aligned on a 32-byte boundary and the table cannot cross a 64 KB boundary in memory.

Physical Region Descriptor Format

Each physical memory region to be transferred is described by a Physical Region Descriptor (PRD) as illustrated in Table 5-11. When the bus master is enabled (Command register bit 0 = 1), data transfer proceeds until each PRD in the PRD table has been transferred. The bus master does not cache PRDs.

The PRD table consists of two DWORDs. The first DWORD contains a 32-bit pointer to a buffer to be transferred. The second DWORD contains the size (16 bits) of the buffer and flags (EOT, EOP, JMP). The description of the flags are as follows:

- EOT bit - If set in a PRD, this bit indicates the last entry in the PRD table (bit 31). The last entry in a PRD table must have either the EOT bit or the JMP bit set. A PRD can not have both the JMP and EOT bits set.
- EOP bit - If set in a PRD and the bus master has completed the PRD's transfer, the End of Page bit is set (Status register bit 0 = 1) and an SMI is generated. If a second EOP is reached due to the completion of another PRD before the End of Page bit is cleared, the Bus Master Error bit is set (Status register bit 1 = 1) and the bus master pauses. In this paused condition, reading the Status register clears both the Bus Master Error and the End of Page bits and the bus master continues.
- JMP bit - This PRD is special. If set, the Memory Region Physical Base Address is now the target address of the JMP. The target address must be on a 32-byte boundary so bits[4:0] must be written to 0. There is no data transfer with this PRD. This PRD allows the creation of a looping mechanism. If a PRD table is created with the JMP bit set in the last PRD, the PRD table does not need a PRD with the EOT bit set. A PRD can not have both the JMP and EOT bits set.

Programming Model

The following discussion explains, in steps, how to initiate and maintain a bus master transfer between memory and an audio slave device.

In the steps listed below, the reference to "Example" refers to Figure 5-12 "PRD Table Example" on page 178.

1) Software creates a PRD table in system memory. Each PRD entry is 8 bytes long; consisting of a base address pointer and buffer size. The maximum data that can be transferred from a PRD entry is 64 KB. A PRD table must be aligned on a 32-byte boundary. The last PRD in a PRD table must have the EOT or JMP bit set.

 Example - Assume the data is outbound. There are three PRDs in the example PRD table. The first two PRDs (PRD_1, PRD_2) have only the EOP bit set. The last PRD (PRD_3) has only the JMP bit set. This example creates a PRD loop.

2) Software loads the starting address of the PRD table by programming the PRD Table Address register.

 Example - Program the PRD Table Address register with Address_3.

3) Software must fill the buffers pointed to by the PRDs with audio data. It is not absolutely necessary to fill the buffers; however, the buffer filling process must stay ahead of the buffer emptying. The simplest way to do this is by using the EOP flags to generate an SMI when a PRD is empty.

 Example - Fill Audio Buffer_1 and Audio Buffer_2. The SMI generated by the EOP from the first PRD allows the software to refill Audio Buffer_1. The second SMI refills Audio Buffer_2. The third SMI refills Audio Buffer_1 and so on.

Table 5-11. Physical Region Descriptor Format

	Byte 3								Byte 2								Byte 1								Byte 0							
DWORD	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	Memory Region Base Address [31:1] (Audio Data Buffer)																															0
1	EOT	EOP	JMP	Reserved													Size [15:1]															0

Core Logic Module (Continued)

4) Read the SMI Status register to clear the Bus Master Error and End of Page bits (bits 1 and 0).

 Set the correct direction to the Read or Write Control bit (Command register bit 3). Note that the direction of the data transfer of a particular bus master is fixed and therefore the direction bit must be programmed accordingly. It is assumed that the codec has been properly programmed to receive the audio data.

 Engage the bus master by writing a "1" to the Bus Master Control bit (Command register bit 0).

 The bus master reads the PRD entry pointed to by the PRD Table Address register and increments the address by 08h to point to the next PRD. The transfer begins.

 Example - The bus master is now properly programmed to transfer Audio Buffer_1 to a specific slot(s) in the AC97 interface.

5) The bus master transfers data to/from memory responding to bus master requests from the AC97 interface. At the completion of each PRD, the bus master's next response depends on the settings of the flags in the PRD.

 Example - At the completion of PRD_1 an SMI is generated because the EOP bit is set while the bus master continues on to PRD_2. The address in the PRD Table Address register is incremented by 08h and is now pointing to PRD_3. The SMI Status register is read to clear the End of Page status flag. Since Audio Buffer_1 is now empty, the software can refill it.

 At the completion of PRD_2 an SMI is generated because the EOP bit is set. The bus master then continues on to PRD_3. The address in the PRD Table Address register is incremented by 08h. The DMA SMI Status register is read to clear the End of Page status flag. Since Audio Buffer_2 is now empty, the software can refill it. Audio Buffer_1 has been refilled from the previous SMI.

 PRD_3 has the JMP bit set. This means the bus master uses the address stored in PRD_3 (Address_3) to locate the next PRD. It does not use the address in the PRD Table Address register to get the next PRD. Since Address_3 is the location of PRD_1, the bus master has looped the PRD table.

Stopping the bus master can be accomplished by not reading the SMI Status register End of Page status flag. This leads to a second EOP which causes a Bus Master Error and pauses the bus master. In effect, once a bus master has been enabled it never has to be disabled, just paused. The bus master cannot be disabled unless the bus master has been paused or has reached an EOT.

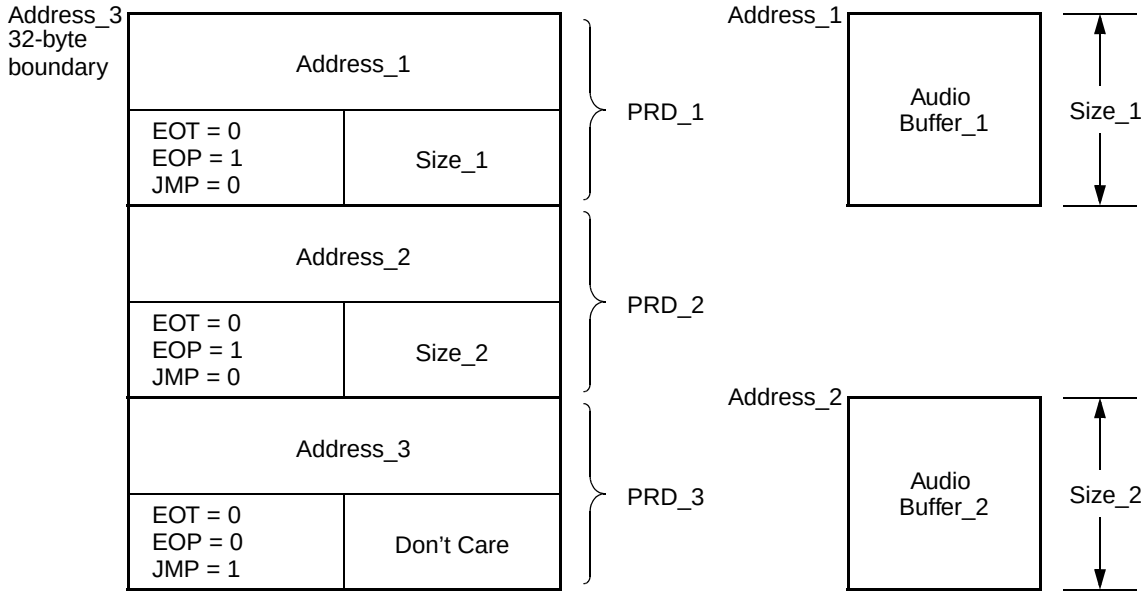


Figure 5-12. PRD Table Example

Core Logic Module (Continued)

5.2.12.2 AC97 Codec Interface

The AC97 codec (e.g., LM4548) is the master of the serial interface and generates the clocks to Core Logic module. Figure 5-13 shows the signal connections between two codecs and the SC2200:

- Codec1 can be AC97 Rev. 1.3 or higher compliant.
- Codec2 is optional, but must be compliant with AC97 2.0 or higher. (For specifics on the serial interface, refer to the appropriate codec manufacturer's datasheet.):
 - — SDATA_IN2 has wakeup capability. (See Section 4.6 "System Wakeup Control (SWC)" on page 125.)
 - — If SDATA_IN2 is not used it must be connected to V_{SS}.
 - — If an AMC97 codec is used (as Codec2), it should be connected to SDATA_IN2 and SDATA_IN should be connected to V_{SS}.
- For PC speaker synthesis, the Core Logic module outputs the PC speaker signal on the PC_BEEP pin which is connected to the PC_BEEP input of the AC97 codec. Note that PC_BEEP is muxed with GPIO16 and must be programmed via PMR[0] (see Table 3-2 on page 81.)

Codec Configuration/Control Registers

The codec 32-bit related registers:

- GPIO Status and Control Registers:
 - — Codec GPIO Status Register (F3BAR0+Memory Offset 00h)
 - — Codec GPIO Control Register (F3BAR0+Memory Offset 04h)
- Codec Status Register (F3BAR0+Memory Offset 08h)
- Codec Command Register (F3BAR0+Memory Offset 0Ch)

Codec GPIO Status and Control Registers:

The Codec GPIO Status and Control registers are used for codec GPIO related tasks such as enabling a codec GPIO interrupt to cause an SMI.

Codec Status Register:

The Codec Status register stores the codec status WORD. It is updated every valid Status Word slot.

Codec Command Register:

The Codec Command register writes the control WORD to the codec. By writing the appropriate control WORDs to this port, the features of the codec can be controlled. The contents of this register are written to the codec during the Control Word slot.

The bit formats for these registers are given in Table 5-38 "F3BAR0+Memory Offset: Audio Configuration Registers" on page 270.

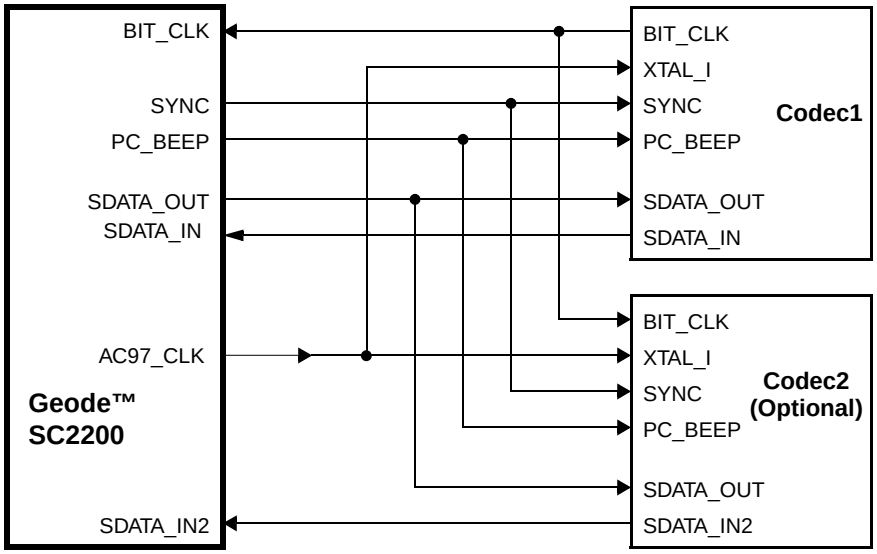


Figure 5-13. AC97 V2.0 Codec Signal Connections

Core Logic Module (Continued)

5.2.12.3 VSA Technology Support Hardware

The Core Logic module incorporates the required hardware in order to support the Virtual System Architecture® (VSA™) technology for capture and playback of audio using an external codec. This eliminates much of the hardware traditionally associated with industry standard audio functions.

VSA Technology

VSA technology provides a framework to enable software implementation of traditionally hardware-only components. VSA software executes in System Management Mode (SMM), enabling it to execute transparently to the operating system, drivers and applications.

The VSA design is based upon a simple model for replacing hardware components with software. Hardware to be virtualized is merely replaced with simple access detection circuitry which asserts the SMI# (System Management Interrupt) internal signal when hardware accesses are detected. The current execution stream is immediately preempted, and the processor enters SMM. The SMM system software then saves the processor state, initializes the VSA execution environment, decodes the SMI source and dispatches handler routines which have registered requests to service the decoded SMI source. Once all handler routines have completed, the processor state is restored and normal execution resumes. In this manner, hardware accesses are transparently replaced with the execution of SMM handler software.

Historically, SMM software was used primarily for the single purpose of facilitating active power management for notebook designs. That software's only function was to manage the power up and down of devices to save power. With high performance processors now available, it is feasible to implement, primarily in SMM software, PC capabilities traditionally provided by hardware. In contrast to power management code, this virtualization software generally has strict performance requirements to prevent application performance from being significantly impacted.

Audio SMI Related Registers

The SMI related registers consist of:

- Audio SMI Status Reporting Registers:
 - — Top Level SMI Mirror and Status Registers (F1BAR0+Memory Offset 00h/02h)
 - — Second Level SMI Status Registers (F3BAR0+Memory Offset 10h/12h)
- I/O Trap SMI and Fast Write Status Register (F3BAR0+Memory Offset 14h)
- I/O Trap SMI Enable Register (F3BAR0+Memory Offset 18h)

Audio SMI Status Reporting Registers:

The Top SMI Status Mirror and Status registers are the top level of hierarchy for the SMI Handler in determining the source of an SMI. These two registers are at F1BAR0+Memory Offset 00h (Status Mirror) and 02h (Status). The registers are identical except that reading the register at F1BAR0+Memory Offset 02h clears the status.

The second level of audio SMI status reporting is set up very much like the top level. There are two status reporting registers, one “read only” (mirror) and one “read to clear”. The data returned by reading either offset is the same (i.e., SMI was caused by an audio related event). The difference between F3BAR0+Memory Offset 10h (Status Mirror) and 12h (Status) is in the ability to clear the SMI source at 12h.

Figure 5-14 on page 181 shows an SMI tree for checking and clearing the source of an audio SMI. Only the audio SMI bit is detailed here. For details regarding the remaining bits in the Top SMI Status Mirror and Status registers refer toTable 5-33 "F1BAR0+I/O Offset: SMI Status Registers" on page 244.

I/O Trap SMI and Fast Write Status Register:

This 32-bit read-only register (F3BAR0+Memory Offset 14h) not only indicates if the enabled I/O trap generated an SMI, but also contains Fast Path Write related bits.

I/O Trap SMI Enable Register:

The I/O Trap SMI Enable register (F3BAR0+Memory Offset 18h) allows traps for specified I/O addresses and configures generation for I/O events. It also contains the enabling bit for Fast Path Read/Write features.

Status Fast Path Read/Write

Status Fast Path Read – If enabled, the Core Logic module intercepts and responds to reads to several status registers. This speeds up operations, and prevents SMI generation for reads to these registers. This process is called Status Fast Path Read. Status Fast Path Read is enabled via F3BAR0+Memory Offset 18h[4].

In Status Fast Path Read the Core Logic module responds to reads of the following addresses:

388h-38Bh, 2x0h, 2x1h, 2x2h, 2x3h, 2x8h and 2x9h

Note that if neither sound card or FM I/O mapping is enabled, then status read trapping is not possible.

Fast Path Write – If enabled, the Core Logic module captures certain writes to several I/O locations. This feature prevents two SMIs from being asserted for write operations that are known to take two accesses (the first access is an index and the second is data). This process is called Fast Path Write. Fast Path Write is enabled in via F3BAR0+Memory Offset 18h[11].

Core Logic Module (Continued)

Fast Path Write captures the data and address bit 1 (A1) of the first access, but does not generate an SMI. A1 is stored in F3BAR0+Memory Offset 14h[15]. The second access causes an SMI, and the data and address are captured as in a normal trapped I/O.

In Fast Path Write, the Core Logic module responds to writes to the following addresses:

388h, 38Ah, 38Bh, 2x0h, 2x2h, and 2x8h

Table 5-38 on page 270 shows the bit formats of the second level SMI status reporting registers and the Fast Path Read/Write programming bits.

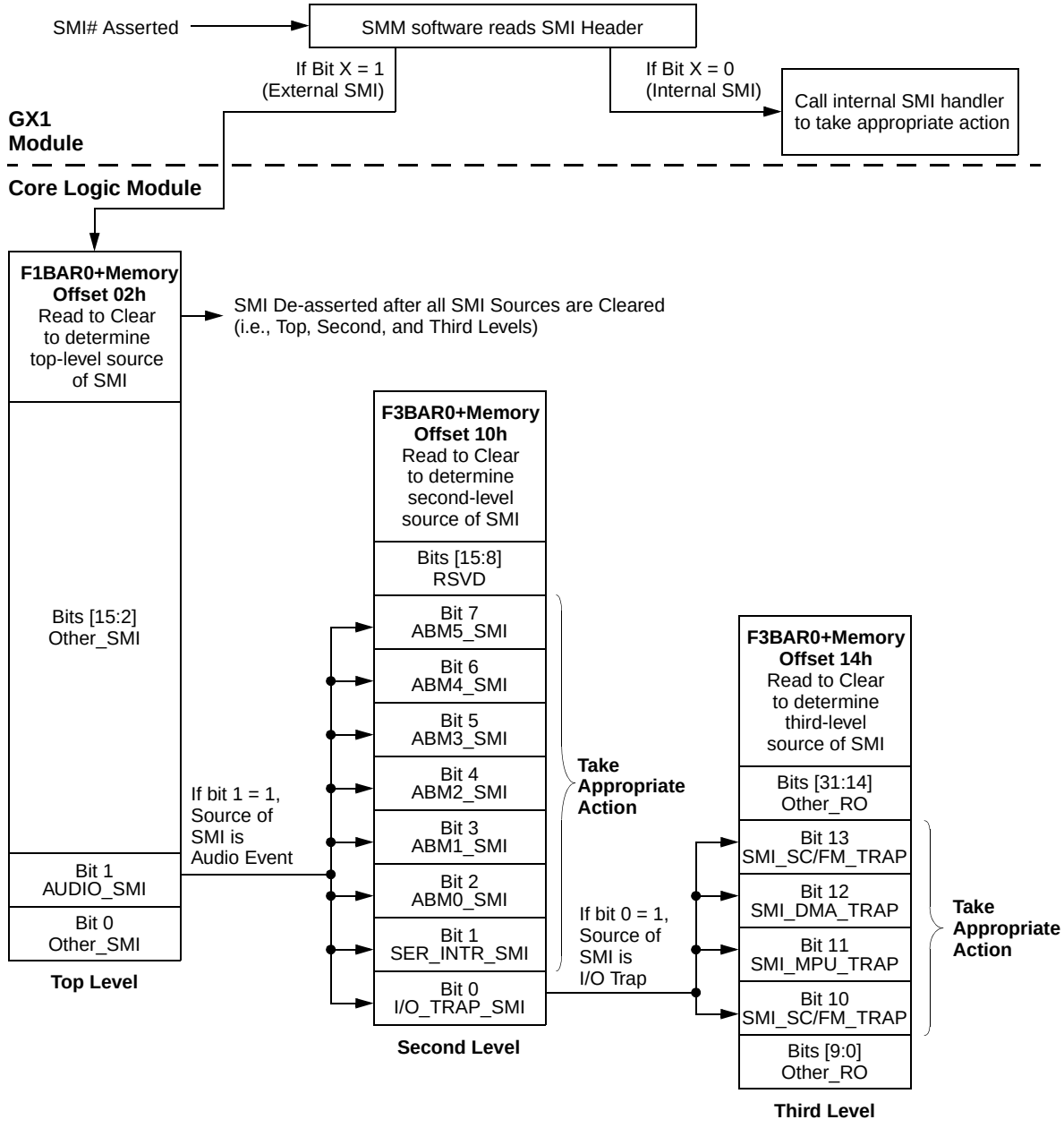


Figure 5-14. Audio SMI Tree Example

Core Logic Module (Continued)

5.2.12.4 IRQ Configuration Registers

The Core Logic module provides the ability to set and clear IRQs internally through software control. If the IRQs are configured for software control, they do not respond to external hardware. There are two registers provided for this feature:

- Internal IRQ Enable Register (F3BAR0+Memory Offset 1Ah)
- Internal IRQ Control Register (F3BAR0+Memory Offset 1Ch)

Internal IRQ Enable Register

The Internal IRQ Enable register configures the IRQs as internal (software) interrupts or external (hardware) interrupts. Any IRQ used as an internal software driven source must be configured as internal.

Internal IRQ Control Register

The Internal IRQ Control register allows individual software assertion/de-assertion of the IRQs that are enabled as internal. These bits are used as masks when attempting to write a particular IRQ bit. If the mask bit is set, it can then be asserted/de-asserted according to the value in the low-order 16 bits. Otherwise the assertion/de-assertion values of the particular IRQ can not be changed.

5.2.12.5 LPC Interface

The LPC interface of the Core Logic module is based on the Intel Low Pin Count (LPC) Interface specification, revision 1.0. In addition to the requirement pins that are specified in the Intel LPC Interface specification, the Core Logic module also supports three optional pins: LDRQ#, SERIRQ, and LPCPD#.

The following subsections briefly describe some sections of the specification. However, for full details refer to the LPC specification directly.

The goals of the LPC interface are to:

- Enable a system without an ISA bus.
- Reduce the cost of traditional ISA bus devices.
- Use on a motherboard only.
- Perform the same cycle types as the ISA bus: memory, I/O, DMA, and Bus Master.
- Increase the memory space from 16 MB to 4 GB to allow BIOS sizes much greater.
- Provide synchronous design. Much of the challenge of an ISA design is meeting the different, and in some cases conflicting, ISA timings. Make the timings synchronous to a reference well known to component designers, such as PCI.
- Support software transparency: do not require special drivers or configuration for this interface. The motherboard BIOS should be able to configure all devices at boot.
- Support desktop and mobile implementations.
- Enable support of a variable number of wait states.
- Enable I/O memory cycle retries in SMM handler.
- Enable support of wakeup and other power state transitions.

Assumptions and functionality requirements of the LPC interface are:

- Only the following class of devices may be connected to the LPC interface:
 - SuperI/O (FDC, SP, PP, IR, KBC) - I/O slave, DMA, bus master (for IR, PP).
 - Audio, including AC97 style design - I/O slave, DMA, bus master.
 - Generic Memory, including BIOS - Memory slave.
 - System Management Controller - I/O slave, bus master.
- Interrupts are communicated with the serial interrupt (SERIRQ) protocol.
- The LPC interface does not need to support high-speed buses (such as CardBus, 1394, etc.) downstream, nor does it need to support low-latency buses such as USB.

Figure 5-15 shows a typical setup. In this setup, the LPC is connected through the Core Logic module to a PCI or host bus.

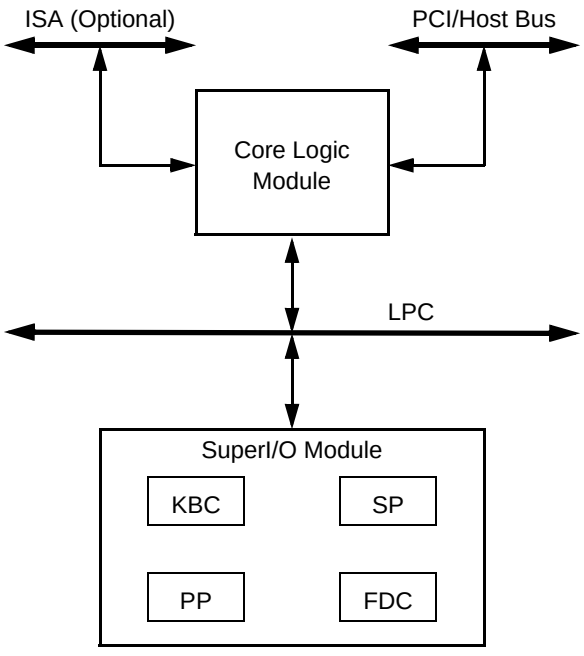


Figure 5-15. Typical Setup

Core Logic Module (Continued)

5.2.12.6 LPC Interface Signal Definitions

The LPC specification lists seven required and six optional signals for supporting the LPC interface. Many of the signals are the same signals found on the PCI interface and do not require any new pins on the host. Required signals must be implemented by both hosts and peripherals. Optional signals may or may not be present on particular hosts or peripherals.

The Core Logic module incorporates all the required LPC interface signals and two of the optional signals:

- Required LPC signals:
 - LAD[3:0] - Multiplexed Command, Address and Data.
 - LFRAME# - Frame: Indicates start of a new cycle, termination of broken cycle.
 - LRESET# - Reset: This signal is not available. Use PCI Reset signal PCIRST# instead.
 - LCLK - Clock: This signal is not available. Use PCI 33 MHz clock signal PCICLK instead.
- Core Logic module optional LPC signals:
 - LDRQ# - Encoded DMA/Bus Master Request: Only needed by peripheral that need DMA or bus mastering. Peripherals may not share the LDRQ# signal.
 - SERIRQ - Serialized IRQ: Only needed by peripherals that need interrupt support.
 - LPCPD# - Power Down: Indicates that the peripheral should prepare for power to the LPC interface to be shut down. Optional for the host.

5.2.12.7 Cycle Types

Table 5-12 shows the various types of cycles that are supported by the Core Logic module.

Table 5-12. Cycle Types

Cycle Type	Supported Sizes (Bytes)
Memory Read	1
Memory Write	1
I/O Read	1
I/O Write	1
DMA Read	1 or 2
DMA Write	1 or 2
Bus Master Memory Read	1, 2, or 4
Bus Master Memory Write	1, 2, or 4

5.2.12.8 LPC Interface Support

The LPC interface supports all the features described in the LPC Bus Interface specification, revision 1.0, with the following exceptions:

- Only 8- or 16-bit DMA, depending on channel number. Does not support the optional larger transfer sizes.
- Only one external DRQ pin.

Core Logic Module (Continued)

5.3 REGISTER DESCRIPTIONS

The Core Logic module is a multi-function module. Its register space can be broadly divided into three categories in which specific types of registers are located:

1) Chipset Register Space (F0-F5) (Note that F4 is for Video Processor support, see Section 6.3.1 on page 335 for register descriptions): Comprised of six separate functions, each with its own register space, consisting of PCI header registers and configuration registers.

 The PCI header is a 256-byte region used for configuring a PCI device or function. The first 64 bytes are the same for all PCI devices and are predefined by the PCI specification. These registers are used to configure the PCI for the device. The rest of the 256-byte region is used to configure the device or function itself.

2) USB Controller Register Space (PCIUSB): Consists of the standard PCI header registers. The USB controller supports three ports and is OpenHCI compliant.

3) ISA Legacy Register Space (I/O Ports): Contains all the legacy compatibility I/O ports that are internal, trapped, shadowed, or snooped.

The following subsections provide:

- A brief discussion on how to access the registers located in PCI Configuration Space.
- Core Logic module register summaries.
- Bit formats for Core Logic module registers.

5.3.1 PCI Configuration Space and Access Methods

Configuration cycles are generated in the processor. All configuration registers in the Core Logic module are accessed through the PCI interface using the PCI Type One Configuration Mechanism. This mechanism uses two DWORD I/O locations at 0CF8h and 0CFCh. The first location (0CF8h) references the Configuration Address register. The second location (0CFCh) references the Configuration Data Register (CDR).

To access PCI configuration space, write the Configuration Address (0CF8h) Register with data that specifies the Core Logic module as the device on PCI being accessed, along with the configuration register offset. On the following cycle, a read or write to the Configuration Data Register (CDR) causes a PCI configuration cycle to the Core Logic module. Byte, WORD, or DWORD accesses are allowed to CDR at 0CFCh, 0CFDh, 0CFEh, or 0CFFh.

The Core Logic module has seven PCI configuration register sets, one for each function (F0-F5) and USB (PCIUSB). Base Address Registers (BARx) in F0-F5 and PCIUSB set the base addresses for additional I/O or memory mapped configuration registers for each function.

Table 5-13 shows the PCI Configuration Address Register (0CF8h) and how to access the PCI header registers.

Table 5-13. PCI Configuration Address Register (0CF8h)

31	30 24	23 16	15 11	10 8	7 2	1 0
Configuration Space Mapping	**Reserved**	**Bus Number**	**Device Number**	**Function**	**Index**	**DWORD 00**
1 (Enable)	000 000	0000 0000	xxxx x **(Note)**	xxx	xxxx xx	00 (Always)
Function 0 (F0): Bridge Configuration, GPIO and LPC Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	000	Index	
Function 1 (F1): SMI Status and ACPI Timer Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	001	Index	
Function 2 (F2): IDE Controller Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	010	Index	
Function 3 (F3): Audio Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	011	Index	
Function 4 (F4): Video Processor Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	100	Index	
Function 5 (F5): X-Bus Expansion Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	101	Index	
PCIUSB: USB Controller Configuration Register Space						
80h		0000 0000	1001 1 or 1000 1	000	Index	

Note: The device number depends upon the IDSEL Strap Override bit (F5BAR0+I/O Offset 04h[0]). This bit allows selection of the address lines to be used as the IDSEL. By Default: IDSEL = AD28 (1001 0) for F0-F5, AD29 (1001 1) for PCIUSB.

Core Logic Module (Continued)

5.3.2 Register Summary

The tables in this subsection summarize the registers of the Core Logic module. Included in the tables are the register's reset values and page references where the bit formats are found.

Note: Function 4 (F4) is for Video Processor support (although accessed through the Core Logic PCI configuration registers). Refer to Section 6.3.1 "Register Summary" on page 335 for details.

Table 5-14. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support Summary

F0 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-29)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 199
02h-03h	16	RO	Device Identification Register	0500h	Page 199
04h-05h	16	R/W	PCI Command Register	000Fh	Page 199
06h-07h	16	R/W	PCI Status Register	0280h	Page 200
08h	8	RO	Device Revision ID Register	00h	Page 200
09h-0Bh	24	RO	PCI Class Code Register	060100h	Page 201
0Ch	8	R/W	PCI Cache Line Size Register	00h	Page 201
0Dh	8	R/W	PCI Latency Timer Register	00h	Page 201
0Eh	8	RO	PCI Header Type Register	80h	Page 201
0Fh	8	RO	PCI BIST Register	00h	Page 201
10h-13h	32	R/W	Base Address Register 0 (F0BAR0) — Sets the base address for the I/O mapped GPIO Runtime and Configuration Registers (summarized in Table 5-15).	00000001h	Page 201
14h-17h	32	R/W	Base Address Register 1 (F0BAR1) — Sets the base address for the I/O mapped LPC Configuration Registers (summarized in Table 5-16)	00000001h	Page 201
18h-2Bh	---	---	Reserved	00h	Page 201
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 201
2Eh-2Fh	16	RO	Subsystem ID	0500h	Page 201
30h-3Fh	---	---	Reserved	00h	Page 201
40h	8	R/W	PCI Function Control Register 1	39h	Page 202
41h	8	R/W	PCI Function Control Register 2	00h	Page 202
42h	---	---	Reserved	00h	Page 203
43h	8	R/W	PIT Delayed Transactions Register	02h	Page 203
44h	8	R/W	Reset Control Register	01h	Page 203
45h	---	---	Reserved	00h	Page 204
46h	8	R/W	PCI Functions Enable Register	FEh	Page 204
47h	8	R/W	Miscellaneous Enable Register	00h	Page 204
48h-4Bh	---	---	Reserved	00h	Page 204
4Ch-4Fh	32	R/W	Top of System Memory	FFFFFFFFh	Page 204
50h	8	R/W	PIT Control/ISA CLK Divider	7Bh	Page 204
51h	8	R/W	ISA I/O Recovery Control Register	40h	Page 205
52h	8	R/W	ROM/AT Logic Control Register	98h	Page 205
53h	8	R/W	Alternate CPU Support Register	00h	Page 206
54h-59h	---	---	Reserved	00h	Page 206
5Ah	8	R/W	Decode Control Register 1	01h	Page 206
5Bh	8	R/W	Decode Control Register 2	20h	Page 207
5Ch	8	R/W	PCI Interrupt Steering Register 1	00h	Page 208
5Dh	8	R/W	PCI Interrupt Steering Register 2	00h	Page 208
5Eh-5Fh	---	---	Reserved	00h	Page 208
60h-63h	32	R/W	ACPI Control Register	00000000h	Page 208
64h-6Dh	---	---	Reserved	00h	Page 209
6Eh-6Fh	16	R/W	ROM Mask Register	FFF0h	Page 209

Core Logic Module (Continued)

Table 5-14. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support Summary (Continued)

F0 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-29)
70h-71h	16	R/W	IOCS1# Base Address Register	0000h	Page 209
72h	8	R/W	IOCS1# Control Register	00h	Page 209
73h	8	---	Reserved	00h	Page 209
74h-75h	16	R/W	IOCS0 Base Address Register	0000h	Page 209
76h	8	R/W	IOCS0 Control Register	00h	Page 210
77h	---	---	Reserved	00h	Page 210
78h-7Bh	32	R/W	DOCCS Base Address Register	00000000h	Page 210
7Ch-7Fh	32	R/W	DOCCS Control Register	00000000h	Page 210
80h	8	R/W	Power Management Enable Register 1	00h	Page 210
81h	8	R/W	Power Management Enable Register 2	00h	Page 211
82h	8	R/W	Power Management Enable Register 3	00h	Page 213
83h	8	R/W	Power Management Enable Register 4	00h	Page 214
84h	8	RO	Second Level PME/SMI Status Mirror Register 1	00h	Page 215
85h	8	RO	Second Level PME/SMI Status Mirror Register 2	00h	Page 215
86h	8	RO	Second Level PME/SMI Status Mirror Register 3	00h	Page 216
87h	8	RO	Second Level PME/SMI Status Mirror Register 4	00h	Page 217
88h	8	R/W	General Purpose Timer 1 Count Register	00h	Page 218
89h	8	R/W	General Purpose Timer 1 Control Register	00h	Page 218
8Ah	8	R/W	General Purpose Timer 2 Count Register	00h	Page 219
8Bh	8	R/W	General Purpose Timer 2 Control Register	00h	Page 219
8Ch	8	R/W	IRQ Speedup Timer Count Register	00h	Page 220
8Dh	8	R/W	Video Speedup Timer Count Register	00h	Page 220
8Eh	8	R/W	VGA Timer Count Register	00h	Page 220
8Fh-92h	---	---	Reserved	00h	Page 220
93h	8	R/W	Miscellaneous Device Control Register	00h	Page 220
94h-95h	16	R/W	Suspend Modulation Register	0000h	Page 221
96h	8	R/W	Suspend Configuration Register	00h	Page 221
97h	---	---	Reserved	00h	Page 221
98h-99h	16	R/W	Hard Disk Idle Timer Count Register — Primary Channel	0000h	Page 222
9Ah-9Bh	16	R/W	Floppy Disk Idle Timer Count Register	0000h	Page 222
9Ch-9Dh	16	R/W	Parallel / Serial Idle Timer Count Register	0000h	Page 222
9Eh-9Fh	16	R/W	Keyboard / Mouse Idle Timer Count Register	0000h	Page 222
A0h-A1h	16	R/W	User Defined Device 1 Idle Timer Count Register	0000h	Page 222
A2h-A3h	16	R/W	User Defined Device 2 Idle Timer Count Register	0000h	Page 222
A4h-A5h	16	R/W	User Defined Device 3 Idle Timer Count Register	0000h	Page 223
A6h-A7h	16	R/W	Video Idle Timer Count Register	0000h	Page 223
A8h-A9h	16	R/W	Video Overflow Count Register	0000h	Page 223
AAh-ABh	---	---	Reserved	00h	Page 223
ACh-ADh	16	R/W	Hard Disk Idle Timer Count Register — Secondary Channel	0000h	Page 223
AEh	8	WO	CPU Suspend Command Register	00h	Page 223
AFh	8	WO	Suspend Notebook Command Register	00h	Page 223
B0h-B3h	---	---	Reserved	00h	Page 223
B4h	8	RO	Floppy Port 3F2h Shadow Register	xxh	Page 224
B5h	8	RO	Floppy Port 3F7h Shadow Register	xxh	Page 224
B6h	8	RO	Floppy Port 1F2h Shadow Register	xxh	Page 224
B7h	8	RO	Floppy Port 1F7h Shadow Register	xxh	Page 224
B8h	8	RO	DMA Shadow Register	xxh	Page 224
B9h	8	RO	PIC Shadow Register	xxh	Page 224

Core Logic Module (Continued)

Table 5-14. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support Summary (Continued)

F0 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-29)
BAh	8	RO	PIT Shadow Register	xxh	Page 225
BBh	8	RO	RTC Index Shadow Register	xxh	Page 225
BCh	8	R/W	Clock Stop Control Register	00h	Page 225
BDh-BFh	---	---	Reserved	00h	Page 225
C0h-C3h	32	R/W	User Defined Device 1 Base Address Register	00000000h	Page 225
C4h-C7h	32	R/W	User Defined Device 2 Base Address Register	00000000h	Page 226
C8h-CBh	32	R/W	User Defined Device 3 Base Address Register	00000000h	Page 226
CCh	8	R/W	User Defined Device 1 Control Register	00h	Page 226
CDh	8	R/W	User Defined Device 2 Control Register	00h	Page 226
CEh	8	R/W	User Defined Device 3 Control Register	00h	Page 227
CFh	---	---	Reserved	00h	Page 227
D0h	8	WO	Software SMI Register	00h	Page 227
D1h-EBh	16	---	Reserved	00h	Page 227
ECh	8	R/W	Timer Test Register	00h	Page 227
EDh-F3h	---	---	Reserved	00h	Page 227
F4h	8	RC	Second Level PME/SMI Status Register 1	00h	Page 227
F5h	8	RC	Second Level PME/SMI Status Register 2	00h	Page 228
F6h	8	RC	Second Level PME/SMI Status Register 3	00h	Page 229
F7h	8	RC	Second Level PME/SMI Status Register 4	00h	Page 230
F8h-FFh	---	---	Reserved	00h	Page 230

Core Logic Module (Continued)

Table 5-15. F0BAR0: GPIO Support Registers Summary

F0BAR0+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-30)
00h-03h	32	R/W	GPDO0 — GPIO Data Out 0 Register	FFFFFFFFh	Page 231
04h-07h	32	RO	GPDI0 — GPIO Data In 0 Register	FFFFFFFFh	Page 231
08h-0Bh	32	R/W	GPIEN0 — GPIO Interrupt Enable 0 Register	00000000h	Page 231
0Ch-0Fh	32	R/W1C	GPST0 — GPIO Status 0 Register	00000000h	Page 231
10h-13h	32	R/W	GPDO1 — GPIO Data Out 1 Register	FFFFFFFFh	Page 231
14h-17h	32	RO	GPDI1 — GPIO Data In 1 Register	FFFFFFFFh	Page 232
18h-1Bh	32	R/W	GPIEN1 — GPIO Interrupt Enable 1 Register	00000000h	Page 232
1Ch-1Fh	32	R/W1C	GPST1 — GPIO Status 1 Register	00000000h	Page 232
20h-23h	32	R/W	GPIO Signal Configuration Select Register	00000000h	Page 232
24h-27h	32	R/W	GPIO Signal Configuration Access Register	00000044h	Page 233
28h-2Bh	32	R/W	GPIO Reset Control Register	00000000h	Page 234

Table 5-16. F0BAR1: LPC Support Registers Summary

F0BAR1+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-31)
00h-03h	32	R/W	SERIRQ_SRC — Serial IRQ Source Register	00000000h	Page 235
04h-07h	32	R/W	SERIRQ_LVL — Serial IRQ Level Control Register	00000000h	Page 236
08h-0Bh	32	R/W	SERIRQ_CNT — Serial IRQ Control Register	00000000h	Page 238
0Ch-0Fh	32	R/W	DRQ_SRC — DRQ Source Register	00000000h	Page 238
10h-13h	32	R/W	LAD_EN — LPC Address Enable Register	00000000h	Page 239
14h-17h	32	R/W	LAD_D0 — LPC Address Decode 0 Register	00080020h	Page 240
18h-1Bh	32	R/W	LAD_D1 — LPC Address Decode 1 Register	00000000h	Page 240
1Ch-1Fh	32	R/W	LPC_ERR_SMI — LPC Error SMI Register	00000080h	Page 241
20h-23h	32	RO	LPC_ERR_ADD — LPC Error Address Register	00000000h	Page 242

Core Logic Module (Continued)

Table 5-17. F1: PCI Header Registers for SMI Status and ACPI Support Summary

F1 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-32)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 243
02h-03h	16	RO	Device Identification Register	0501h	Page 243
04h-05h	16	R/W	PCI Command Register	0000h	Page 243
06h-07h	16	RO	PCI Status Register	0280h	Page 243
08h	8	RO	Device Revision ID Register	00h	Page 243
09h-0Bh	24	RO	PCI Class Code Register	068000h	Page 243
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 243
0Dh	8	RO	PCI Latency Timer Register	00h	Page 243
0Eh	8	RO	PCI Header Type Register	00h	Page 243
0Fh	8	RO	PCI BIST Register	00h	Page 243
10h-13h	32	R/W	Base Address Register 0 (F1BAR0) — Sets the base address for the I/O mapped SMI Status Registers (summarized in Table 5-18).	00000001h	Page 243
14h-2Bh	---	---	Reserved	00h	Page 243
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 243
2Eh-2Fh	16	RO	Subsystem ID	0501h	Page 243
30h-3Fh	---	---	Reserved	00h	Page 243
40h-43h	32	R/W	Base Address Register 1 (F1BAR1) — Sets the base address for the I/O mapped ACPI Support Registers (summarized in Table 5-19)	00000001h	Page 243
44h-FFh	---	---	Reserved	00h	Page 243

Table 5-18. F1BAR0: SMI Status Registers Summary

F1BAR0+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-33)
00h-01h	16	RO	Top Level PME/SMI Status Mirror Register	0000h	Page 244
02h-03h	16	RO/RC	Top Level PME/SMI Status Register	0000h	Page 245
04h-05h	16	RO	Second Level General Traps & Timers PME/SMI Status Mirror Register	0000h	Page 247
06h-07h	16	RC	Second Level General Traps & Timers PME/SMI Status Register	0000h	Page 248
08h-09h	16	Read to Enable	SMI Speedup Disable Register	0000h	Page 248
0Ah-1Bh	---	---	Reserved	00h	Page 248
1Ch-1Fh	32	RO	ACPI Timer Register	xxxxxxxxh	Page 248
20h-21h	16	RO	Second Level ACPI PME/SMI Status Mirror Register	0000h	Page 249
22h-23h	16	RC	Second Level ACPI PME/SMI Status Register	0000h	Page 249
24h-27h	32	R/W	External SMI Register	00000000h	Page 250
28h-4Fh	---	---	Not Used	00h	Page 252
50h-FFh	---	---	The I/O mapped registers located here (F1BAR0+I/O Offset 50h-FFh) are also accessible at F0 Index 50h-FFh. The preferred method is to program these registers through the F0 register space.		

Core Logic Module (Continued)

Table 5-19. F1BAR1: ACPI Support Registers Summary

F1BAR1+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-34)
00h-03h	32	R/W	P_CNT — Processor Control Register	00000000h	Page 253
04h	8	RO	Reserved, do not read	00h	Page 253
05h	8	RO	P_LVL3 — Enter C3 Power State Register	xxh	Page 253
06h	8	R/W	SMI_CMD — OS/BIOS Requests Register	00h	Page 253
07h	8	R/W	ACPI_FUN_CNT — ACPI Function Control Register	00h	Page 253
08h-09h	16	R/W	PM1A_STS — PM1A Status Register	0000h	Page 254
0Ah-0Bh	16	R/W	PM1A_EN — PM1A Enable Register	0000h	Page 255
0Ch-0Dh	16	R/W	PM1A_CNT — PM1A Control Register	0000h	Page 256
0Eh	8	R/W	ACPI_BIOS_STS Register	00h	Page 256
0Fh	8	R/W	ACPI_BIOS_EN Register	00h	Page 257
10h-11h	16	R/W	GPE0_STS — General Purpose Event 0 Status Register	xxxxh	Page 257
12h-13h	16	R/W	GPE0_EN — General Purpose Event 0 Enable Register	0000h	Page 258
14h	8	R/W	GPWIO Control Register 1	00h	Page 259
15h	8	R/W	GPWIO Control Register 2	00h	Page 260
16h	8	R/W	GPWIO Data Register	00h	Page 261
17h	---	---	Reserved	00h	Page 261
18h-1Bh	32	R/W	ACPI SCI_ROUTING Register	00000F00h	Page 261
1Ch-1Fh	32	RO	PM_TMR — ACPI Timer Register	xxxxxxxxh	Page 262
20h	8	R/W	PM2_CNT — PM2 Control Register	00h	Page 262
21h-FFh	---	---	Not Used	00h	Page 262

Table 5-20. F2: PCI Header Registers for IDE Controller Support Summary

F2 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-35)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 263
02h-03h	16	RO	Device Identification Register	0502h	Page 263
04h-05h	16	R/W	PCI Command Register	0000h	Page 263
06h-07h	16	RO	PCI Status Register	0280h	Page 263
08h	8	RO	Device Revision ID Register	01h	Page 263
09h-0Bh	24	RO	PCI Class Code Register	010180h	Page 263
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 263
0Dh	8	RO	PCI Latency Timer Register	00h	Page 263
0Eh	8	RO	PCI Header Type Register	00h	Page 263
0Fh	8	RO	PCI BIST Register	00h	Page 263
10h-13h	32	RO	Base Address Register 0 (F2BAR0) — Reserved for possible future use by the Core Logic module.	00000000h	Page 263
14h-17h	32	RO	Base Address Register 1 (F2BAR1) — Reserved for possible future use by the Core Logic module.	00000000h	Page 263
18h-1Bh	32	RO	Base Address Register 2 (F2BAR2) — Reserved for possible future use by the Core Logic module.	00000000h	Page 263
1Ch-1Fh	32	RO	Base Address Register 3 (F2BAR3) — Reserved for possible future use by the Core Logic module.	00000000h	Page 263
20h-23h	32	R/W	Base Address Register 4 (F2BAR4) — Sets the base address for the I/O mapped Bus Master IDE Registers (summarized in Table 5-21)	00000001h	Page 263
24h-2Bh	---	---	Reserved	00h	Page 263
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 263
2Eh-2Fh	16	RO	Subsystem ID	0502h	Page 263
30h-3Fh	---	---	Reserved	00h	Page 263
40h-43h	32	R/W	Channel 0 Drive 0 PIO Register	00009172h	Page 264
44h-47h	32	R/W	Channel 0 Drive 0 DMA Control Register	00077771h	Page 265
48h-4Bh	32	R/W	Channel 0 Drive 1 PIO Register	00009172h	Page 266
4Ch-4Fh	32	R/W	Channel 0 Drive 1 DMA Control Register	00077771h	Page 266
50h-53h	32	R/W	Channel 1 Drive 0 PIO Register	00009172h	Page 266
54h-57h	32	R/W	Channel 1 Drive 0 DMA Control Register	00077771h	Page 266
58h-5Bh	32	R/W	Channel 1 Drive 1 PIO Register	00009172h	Page 266
5Ch-5Fh	32	R/W	Channel 1 Drive 1 DMA Control Register	00077771h	Page 266
60h-FFh	---	---	Reserved	00h	Page 266

Core Logic Module (Continued)

Table 5-21. F2BAR4: IDE Controller Support Registers Summary

F2BAR4+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-36)
00h	8	R/W	IDE Bus Master 0 Command Register — Primary	00h	Page 267
01h	---	---	Not Used	---	Page 267
02h	8	R/W	IDE Bus Master 0 Status Register — Primary	00h	Page 267
03h	---	---	Not Used	---	Page 267
04h-07h	32	R/W	IDE Bus Master 0 PRD Table Address — Primary	00000000h	Page 267
08h	8	R/W	IDE Bus Master 1 Command Register — Secondary	00h	Page 268
09h	---	---	Not Used	---	Page 268
0Ah	8	R/W	IDE Bus Master 1 Status Register — Secondary	00h	Page 268
0Bh	---	---	Not Used	---	Page 268
0Ch-0Fh	32	R/W	IDE Bus Master 1 PRD Table Address — Secondary	00000000h	Page 268

Table 5-22. F3: PCI Header Registers for Audio Support Summary

F3 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-37)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 269
02h-03h	16	RO	Device Identification Register	0503h	Page 269
04h-05h	16	R/W	PCI Command Register	0000h	Page 269
06h-07h	16	RO	PCI Status Register	0280h	Page 269
08h	8	RO	Device Revision ID Register	00h	Page 269
09h-0Bh	24	RO	PCI Class Code Register	040100h	Page 269
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 269
0Dh	8	RO	PCI Latency Timer Register	00h	Page 269
0Eh	8	RO	PCI Header Type Register	00h	Page 269
0Fh	8	RO	PCI BIST Register	00h	Page 269
10h-13h	32	R/W	Base Address Register 0 (F3BAR0) — Sets the base address for the memory mapped VSA audio interface control register block (summarized in Table 5-23).	00000000h	Page 269
14h-2Bh	---	---	Reserved	00h	Page 269
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 269
2Eh-2Fh	16	RO	Subsystem ID	0503h	Page 269
30h-FFh	---	---	Reserved	00h	Page 269

Core Logic Module (Continued)

Table 5-23. F3BAR0: Audio Support Registers Summary

F3BAR0+ Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-38)
00h-03h	32	R/W	Codec GPIO Status Register	00000000h	Page 270
04h-07h	32	R/W	Codec GPIO Control Register	00000000h	Page 270
08h-0Bh	32	R/W	Codec Status Register	00000000h	Page 270
0Ch-0Fh	32	R/W	Codec Command Register	00000000h	Page 271
10h-11h	16	RC	Second Level Audio SMI Status Register	0000h	Page 271
12h-13h	16	RO	Second Level Audio SMI Status Mirror Register	0000h	Page 272
14h-17h	32	RO	I/O Trap SMI and Fast Write Status Register	00000000h	Page 273
18h-19h	16	R/W	I/O Trap SMI Enable Register	0000h	Page 274
1Ah-1Bh	16	R/W	Internal IRQ Enable Register	0000h	Page 275
1Ch-1Fh	32	R/W	Internal IRQ Control Register	00000000h	Page 276
20h	8	R/W	Audio Bus Master 0 Command Register	00h	Page 277
21h	8	RC	Audio Bus Master 0 SMI Status Register	00h	Page 278
22h-23h	---	---	Not Used	---	Page 278
24h-27h	32	R/W	Audio Bus Master 0 PRD Table Address	00000000h	Page 278
28h	8	R/W	Audio Bus Master 1 Command Register	00h	Page 278
29h	8	RC	Audio Bus Master 1 SMI Status Register	00h	Page 279
2Ah-2Bh	---	---	Not Used	---	Page 279
2Ch-2Fh	32	R/W	Audio Bus Master 1 PRD Table Address	00000000h	Page 279
30h	8	R/W	Audio Bus Master 2 Command Register	00h	Page 279
31h	8	RC	Audio Bus Master 2 SMI Status Register	00h	Page 280
32h-33h	---	---	Not Used	00h	Page 280
34h-37h	32	R/W	Audio Bus Master 2 PRD Table Address	00000000h	Page 280
38h	8	R/W	Audio Bus Master 3 Command Register	00h	Page 280
39h	8	RC	Audio Bus Master 3 SMI Status Register	00h	Page 281
3Ah-3Bh	---	---	Not Used	---	Page 281
3Ch-3Fh	32	R/W	Audio Bus Master 3 PRD Table Address	00000000h	Page 281
40h	8	R/W	Audio Bus Master 4 Command Register	00h	Page 281
41h	8	RC	Audio Bus Master 4 SMI Status Register	00h	Page 282
42h-43h	---	---	Not Used	---	Page 282
44h-47h	32	R/W	Audio Bus Master 4 PRD Table Address	00000000h	Page 282
48h	8	R/W	Audio Bus Master 5 Command Register	00h	Page 282
49h	8	RC	Audio Bus Master 5 SMI Status Register	00h	Page 283
4Ah-4Bh	---	---	Not Used	---	Page 283
4Ch-4Fh	32	R/W	Audio Bus Master 5 PRD Table Address	00000000h	Page 283

Core Logic Module (Continued)

Table 5-24. F5: PCI Header Registers for X-Bus Expansion Support Summary

F5 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-39)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 284
02h-03h	16	RO	Device Identification Register	0505h	Page 284
04h-05h	16	R/W	PCI Command Register	0000h	Page 284
06h-07h	16	RO	PCI Status Register	0280h	Page 284
08h	8	RO	Device Revision ID Register	00h	Page 284
09h-0Bh	24	RO	PCI Class Code Register	068000h	Page 284
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 284
0Dh	8	RO	PCI Latency Timer Register	00h	Page 284
0Eh	8	RO	PCI Header Type Register	00h	Page 284
0Fh	8	RO	PCI BIST Register	00h	Page 284
10h-13h	32	R/W	Base Address Register 0 (F5BAR0) — Sets the base address for the X-Bus Expansion support registers (summarized in Table 5-25.)	00000000h	Page 284
14h-17h	32	R/W	Base Address Register 1 (F5BAR1) — Reserved for possible future use by the Core Logic module.	00000000h	Page 284
18h-1Bh	32	R/W	Base Address Register 2 (F5BAR2) — Reserved for possible future use by the Core Logic module.	00000000h	Page 284
1Ch-1Fh	32	R/W	Base Address Register 3 (F5BAR3) — Reserved for possible future use by the Core Logic module.	00000000h	Page 284
20h-23h	32	R/W	Base Address Register 4 (F5BAR4) — Reserved for possible future use by the Core Logic module.	00000000h	Page 285
24h-27h	32	R/W	Base Address Register 5 (F5BAR5) — Reserved for possible future use by the Core Logic module.	00000000h	Page 285
28h-2Bh	---	---	Reserved	00h	Page 285
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 285
2Eh-2Fh	16	RO	Subsystem ID	0505h	Page 285
30h-3Fh	---	---	Reserved	00h	Page 285
40h-43h	32	R/W	F5BAR0 Base Address Register Mask	FFFFFFC1h	Page 285
44h-47h	32	R/W	F5BAR1 Base Address Register Mask	00000000h	Page 285
48h-4Bh	32	R/W	F5BAR2 Base Address Register Mask	00000000h	Page 286
4Ch-4Fh	32	R/W	F5BAR3 Base Address Register Mask	00000000h	Page 286
50h-53h	32	R/W	F5BAR4 Base Address Register Mask	00000000h	Page 286
54h-57h	32	R/W	F5BAR5 Base Address Register Mask	00000000h	Page 286
58h	8	R/W	F5BARx Initialized Register	00h	Page 286
59h-FFh	---	---	Reserved	xxh	Page 286
60h-63h	32	R/W	Scratchpad for Chip Number	00000000h	Page 286
64h-67h	32	R/W	Scratchpad for Configuration Block Address	00000000h	Page 286
68h-FFh	---	---	Reserved	---	Page 286

Table 5-25. F5BAR0: I/O Control Support Registers Summary

F5BAR0+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-40)
00h-03h	32	R/W	I/O Control Register 1	010C0007h	Page 287
04h-07h	32	R/W	I/O Control Register 2	00000002h	Page 287
08h-0Bh	32	R/W	I/O Control Register 3	00009000h	Page 288

Table 5-26. PCIUSB: USB PCI Configuration Register Summary

PCIUSB Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-41)
00h-01h	16	RO	Vendor Identification	0E11h	Page 289
02h-03h	16	RO	Device Identification	A0F8h	Page 289
04h-05h	16	R/W	Command Register	00h	Page 289
06h-07h	16	R/W	Status Register	0280h	Page 290
08h	8	RO	Device Revision ID	08h	Page 290
09h-0Bh	24	RO	Class Code	0C0310h	Page 290
0Ch	8	R/W	Cache Line Size	00h	Page 290
0Dh	8	R/W	Latency Timer	00h	Page 290
0Eh	8	RO	Header Type	00h	Page 290
0Fh	8	RO	BIST Register	00h	Page 290
10h-13h	32	R/W	Base Address 0	00000000h	Page 290
14h-2Bh	---	---	Reserved	00h	Page 291
2Ch-2Dh	16	RO	Subsystem Vendor ID	0E11h	Page 291
2Eh-2Fh	16	RO	Subsystem ID	A0F8h	Page 291
30h-3Bh	---	---	Reserved	00h	Page 291
3Ch	8	R/W	Interrupt Line Register	00h	Page 291
3Dh	8	R/W	Interrupt Pin Register	01h	Page 291
3Eh	8	RO	Min. Grant Register	00h	Page 291
3Fh	8	RO	Max. Latency Register	50h	Page 291
40h-43h	32	R/W	ASIC Test Mode Enable Register	000F0000h	Page 291
44h	8	R/W	ASIC Operational Mode Enable	00h	Page 291
45h-FFh	---	---	Reserved	00h	Page 291

Table 5-27. USB_BAR: USB Controller Registers Summary

USB_BAR0 +Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-42)
00h-03h	32	R/W	HcRevision	00000110h	Page 292
04h-07h	32	R/W	HcControl	00000000h	Page 292
08h-0Bh	32	R/W	HcCommandStatus	00000000h	Page 292
0Ch-0Fh	32	R/W	HcInterruptStatus	00000000h	Page 292
10h-13h	32	R/W	HcInterruptEnable	00000000h	Page 293
14h-17h	32	R/W	HcInterruptDisable	00000000h	Page 293
18h-1Bh	32	R/W	HcHCCA	00000000h	Page 294
1Ch-1Fh	32	R/W	HcPeriodCurrentED	00000000h	Page 294
20h-23h	32	R/W	HcControlHeadED	00000000h	Page 294
24h-27h	32	R/W	HcControlCurrentED	00000000h	Page 294
28h-2Bh	32	R/W	HcBulkHeadED	00000000h	Page 294
2Ch-2Fh	32	R/W	HcBulkCurrentED	00000000h	Page 294
30h-33h	32	R/W	HcDoneHead	00000000h	Page 294
34h-37h	32	R/W	HcFmInterval	00002EDFh	Page 294
38h-3Bh	32	RO	HcFrameRemaining	00000000h	Page 295
3Ch-3Fh	32	RO	HcFmNumber	00000000h	Page 295
40h-43h	32	R/W	HcPeriodicStart	00000000h	Page 295
44h-47h	32	R/W	HcLSThreshold	00000628h	Page 295
48h-4Bh	32	R/W	HcRhDescriptorA	01000003h	Page 295
4Ch-4Fh	32	R/W	HcRhDescriptorB	00000000h	Page 296
50h-53h	32	R/W	HcRhStatus	00000000h	Page 296
54h-57h	32	R/W	HcRhPortStatus[1]	00000000h	Page 297
58h-5Bh	32	R/W	HcRhPortStatus[2]	00000000h	Page 298
5Ch-5Fh	32	R/W	HcRhPortStatus[3]	00000000h	Page 299
60h-9Fh	---	---	Reserved	xxxxxxxxh	Page 300
100h-103h	32	R/W	HceControl	00000000h	Page 300
104h-107h	32	R/W	HceInput	000000xxh	Page 301
108h-10Dh	32	R/W	HceOutput	000000xxh	Page 301
10Ch-10Fh	32	R/W	HceStatus	00000000h	Page 301

Table 5-28. ISA Legacy I/O Register Summary

I/O Port	Type	Name	Reference
DMA Channel Control Registers (Table 5-43)			
000h	R/W	DMA Channel 0 Address Register	Page 302
001h	R/W	DMA Channel 0 Transfer Count Register	Page 302
002h	R/W	DMA Channel 1 Address Register	Page 302
003h	R/W	DMA Channel 1 Transfer Count Register	Page 302
004h	R/W	DMA Channel 2 Address Register	Page 302
005h	R/W	DMA Channel 2 Transfer Count Register	Page 302
006h	R/W	DMA Channel 3 Address Register	Page 302
007h	R/W	DMA Channel 3 Transfer Count Register	Page 302
008h	Read	DMA Status Register, Channels 3:0	Page 302
	Write	DMA Command Register, Channels 3:0	Page 303
009h	WO	Software DMA Request Register, Channels 3:0	Page 303
00Ah	W	DMA Channel Mask Register, Channels 3:0	Page 303
00Bh	WO	DMA Channel Mode Register, Channels 3:0	Page 304
00Ch	WO	DMA Clear Byte Pointer Command, Channels 3:0	Page 304
00Dh	WO	DMA Master Clear Command, Channels 3:0	Page 304
00Eh	WO	DMA Clear Mask Register Command, Channels 3:0	Page 304
00Fh	WO	DMA Write Mask Register Command, Channels 3:0	Page 304
0C0h	R/W	DMA Channel 4 Address Register (Not used)	Page 304
0C2h	R/W	DMA Channel 4 Transfer Count Register (Not Used)	Page 304
0C4h	R/W	DMA Channel 5 Address Register	Page 304
0C6h	R/W	DMA Channel 5 Transfer Count Register	Page 304
0C8h	R/W	DMA Channel 6 Address Register	Page 304
0CAh	R/W	DMA Channel 6 Transfer Count Register	Page 304
0CCh	R/W	DMA Channel 7 Address Register	Page 304
0CEh	R/W	DMA Channel 7 Transfer Count Register	Page 305
0D0h	Read	DMA Status Register, Channels 7:4	Page 305
	Write	DMA Command Register, Channels 7:4	Page 305
0D2h	WO	Software DMA Request Register, Channels 7:4	Page 306
0D4h	W	DMA Channel Mask Register, Channels 7:4	Page 306
0D6h	WO	DMA Channel Mode Register, Channels 7:4	Page 306
0D8h	WO	DMA Clear Byte Pointer Command, Channels 7:4	Page 306
0DAh	WO	DMA Master Clear Command, Channels 7:4	Page 306
0DCh	WO	DMA Clear Mask Register Command, Channels 7:4	Page 306
0DEh	WO	DMA Write Mask Register Command, Channels 7:4	Page 306
DMA Page Registers (Table 5-44)			
081h	R/W	DMA Channel 2 Low Page Register	Page 307
082h	R/W	DMA Channel 3 Low Page Register	Page 307
083h	R/W	DMA Channel 1 Low Page Register	Page 307
087h	R/W	DMA Channel 0 Low Page Register	Page 307
089h	R/W	DMA Channel 6 Low Page Register	Page 307
08Ah	R/W	DMA Channel 7 Low Page Register	Page 307
08Bh	R/W	DMA Channel 5 Low Page Register	Page 307
08Fh	R/W	Sub-ISA Refresh Low Page Register	Page 307
481h	R/W	DMA Channel 2 High Page Register	Page 307
482h	R/W	DMA Channel 3 High Page Register	Page 307
483h	R/W	DMA Channel 1 High Page Register	Page 307
487h	R/W	DMA Channel 0 High Page Register	Page 307

Core Logic Module (Continued)

Table 5-28. ISA Legacy I/O Register Summary (Continued)

I/O Port	Type	Name	Reference
489h	R/W	DMA Channel 6 High Page Register	Page 307
48Ah	R/W	DMA Channel 7 High Page Register	Page 307
48Bh	R/W	DMA Channel 5 High Page Register	Page 307
Programmable Interval Timer Registers (Table 5-45)			
040h	W	PIT Timer 0 Counter	Page 308
	R	PIT Timer 0 Status	Page 308
041h	W	PIT Timer 1 Counter (Refresh)	Page 308
	R	PIT Timer 1 Status (Refresh)	Page 308
042h	W	PIT Timer 2 Counter (Speaker)	Page 309
	R	PIT Timer 2 Status (Speaker)	Page 309
043h	R/W	PIT Mode Control Word Register	Page 309
		Read Status Command	
		Counter Latch Command	
Programmable Interrupt Controller Registers (Table 5-46)			
020h / 0A0h	WO	Master / Slave PCI ICW1	Page 310
021h / 0A1h	WO	Master / Slave PIC ICW2	Page 310
021h / 0A1h	WO	Master / Slave PIC ICW3	Page 310
021h / 0A1h	WO	Master / Slave PIC ICW4	Page 310
021h / 0A1h	R/W	Master / Slave PIC OCW1	Page 310
020h / 0A0h	WO	Master / Slave PIC OCW2	Page 311
020h / 0A0h	WO	Master / Slave PIC OCW3	Page 311
020h / 0A0h	RO	Master / Slave PIC Interrupt Request and Service Registers for OCW3 Commands	Page 311
Keyboard Controller Registers (Table 5-47)			
060h	R/W	External Keyboard Controller Data Register	Page 313
061h	R/W	Port B Control Register	Page 313
062h	R/W	External Keyboard Controller Mailbox Register	Page 313
064h	R/W	External Keyboard Controller Command Register	Page 313
066h	R/W	External Keyboard Controller Mailbox Register	Page 313
092h	R/W	Port A Control Register	Page 313
Real-Time Clock Registers (Table 5-48)			
070h	WO	RTC Address Register	Page 314
071h	R/W	RTC Data Register	Page 314
072h	WO	RTC Extended Address Register	Page 314
073h	R/W	RTC Extended Data Register	Page 314
Miscellaneous Registers (Table 5-49)			
0F0h, 0F1h	WO	Coprocessor Error Register	Page 314
170h-177h/ 376h-377h	R/W	Secondary IDE Registers	Page 314
1F0-1F7h/ 3F6h-3F7h	R/W	Primary IDE Registers	Page 314
4D0h	R/W	Interrupt Edge/Level Select Register 1	Page 314
4D1h	R/W	Interrupt Edge/Level Select Register 2	Page 315

5.4 CHIPSET REGISTER SPACE

The Chipset Register Space of the Core Logic module is comprised of six separate functions (F0-F5), each with its own register space. Base Address Registers (BARs) in each PCI header register space set the base address for the configuration registers for each respective function. The configuration registers accessed through BARs are I/O or memory mapped. The PCI header registers in all functions are very similar.

1) Function 0 (F0): PCI Header/Bridge Configuration Registers for GPIO, and LPC Support (see Section 5.4.1).
2) Function 1 (F1): PCI Header Registers for SMI Status and ACPI Support (see Section 5.4.2 on page 243).
3) Function 2 (F2): PCI Header/Channel 0 and 1 Configuration Registers for IDE Controller Support (see Section 5.4.3 on page 263).
4) Function 3 (F3): PCI Header Registers for Audio Support (see Section 5.4.4 on page 269).
5) Function 4 (F4): PCI Header Registers Video Processor Support (see Section 6.3 on page 335).
6) Function 5 (F5): PCI Header Registers for X-Bus Expansion Support (see Section 5.4.5 on page 284).

 Function 5 contain six BARs in their standard PCI header locations (i.e., Index 10h, 14h, 18h, 1Ch, 20h, and 24h). In addition there are six mask registers that allow the six BARs to be fully programmable from 4 GB to 16 bytes for memory and from 4 GB to 4 bytes for I/O

General Remarks:

- Reserved bits that are defined as "must be set to 0 or 1" should be written with that value.
- Reserved bits that are not defined as "must be set to 0 or 1" should be written with a value that is read from them.
- "Read to Clear" registers that are wider than one byte should be read in one read operation. If they are read a byte at a time, status bits may be lost, or not cleared.

5.4.1 Bridge, GPIO, and LPC Registers - Function 0

The register space designated as Function 0 (F0) is used to configure Bridge features and functionality unique to the Core Logic module. In addition, it configures the PCI portion of support hardware for the GPIO and LPC support registers. The bit formats for the PCI Header and Bridge Configuration registers are given in Table 5-29.

Note: The registers at F0 Index 50h-FFh can also be accessed at F1BAR0+I/O Offset 50h-FFh. However, the preferred method is to program these registers through the F0 register space.

Located in the PCI Header registers of F0, are two Base Address Registers (F0BARx) used for pointing to the register spaces designated for GPIO and LPC configuration (described in Section 5.4.1.1 "GPIO Support Registers" on page 231 and Section 5.4.1.2 "LPC Support Registers" on page 235).

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0500h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 000Fh**
15:10	**Reserved.** Must be set to 0.
9	**Fast Back-to-Back Enable.** This function is not supported when the Core Logic module is a master. It must always be disabled (i.e., must be set to 0).
8	SERR#. Allow SERR# assertion on detection of special errors. 0: Disable. (Default) 1: Enable.
7	**Wait Cycle Control (Read Only).** This function is not supported in the Core Logic module. It is always disabled (always reads 0, hardwired).
6	**Parity Error.** Allow the Core Logic module to check for parity errors on PCI cycles for which it is a target and to assert PERR# when a parity error is detected. 0: Disable. (Default) 1: Enable.
5	**VGA Palette Snoop Enable. (Read Only)** This function is not supported in the Core Logic module. It is always disabled (always reads 0, hardwired).

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4	**Memory Write and Invalidate.** Allow the Core Logic module to do memory write and invalidate cycles, if the PCI Cache Line register (F0 Index 0Ch) is set to 32 bytes (08h). 0: Disable. (Default) 1: Enable.
3	**Special Cycles.** Allow the Core Logic module to respond to special cycles. 0: Disable. 1: Enable. (Default) This bit must be enabled to allow the internal CPU Warm Reset signal to be triggered from a CPU Shutdown cycle.
2	**Bus Master.** Allow the Core Logic module bus mastering capabilities. 0: Disable. 1: Enable. (Default) This bit must be set to 1.
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. (Default)
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycles from the PCI bus: 0: Disable. 1: Enable. (Default) This bit must be set to 1 to access I/O offsets through F0BAR0 and F0BAR1 (see F0 Index 10h and 14h).
Index 06h-07h	**PCI Status Register (R/W)** — **Reset Value: 0280h**
15	**Detected Parity Error.** This bit is set whenever a parity error is detected. Write 1 to clear.
14	**Signaled System Error.** This bit is set whenever the Core Logic module asserts SERR# active. Write 1 to clear.
13	**Received Master Abort.** This bit is set whenever a master abort cycle occurs. A master abort occurs when a PCI cycle is not claimed, except for special cycles. Write 1 to clear.
12	**Received Target Abort.** This bit is set whenever a target abort is received while the Core Logic module is the master for the PCI cycle. Write 1 to clear.
11	**Signaled Target Abort.** This bit is set whenever the Core Logic module signals a target abort. This occurs when an address parity error occurs for an address that hits in the active address decode space of the Core Logic module. Write 1 to clear.
10:9	**DEVSEL# Timing. (Read Only)** These bits are always 01, as the Core Logic module always responds to cycles for which it is an active target with medium DEVSEL# timing. 00: Fast. 01: Medium. 10: Slow. 11: Reserved.
8	**Data Parity Detected.** This bit is set when: 1) The Core Logic module asserts PERR# or observed PERR# asserted. 2) The Core Logic module is the master for the cycle in which the PERR# occurred, and PE is set (F0 Index 04h[6] = 1). Write 1 to clear.
7	**Fast Back-to-Back Capable. (Read Only)** Enables the Core Logic module, as a target, to accept fast back-to-back transactions. 0: Disable. 1: Enable. This bit is always set to 1.
6:0	**Reserved. (Read Only)** Must be set to 0 for future use.
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 060100h**
Index 0Ch	**PCI Cache Line Size Register (R/W)** — **Reset Value: 00h**
7:0	**PCI Cache Line Size Register.** This register sets the size of the PCI cache line, in increments of four bytes. For memory write and invalidate cycles, the PCI cache line size must be set to 32 bytes (08h) and the Memory Write and Invalidate bit (F0 Index 04h[4]) must be set to 1.
Index 0Dh	**PCI Latency Timer Register (R/W)** — **Reset Value: 00h**
7:4	**Reserved.** Must be set to 0.
3:0	**PCI Latency Timer Value.** The PCI Latency Timer register prevents system lockup when a slave does not respond to a cycle that the Core Logic module masters. If the value is set to 00h (default), the timer is disabled. If the timer is written with any other value, bits [3:0] become the four most significant bits in a timer that counts PCI clocks for slave response. The timer is reset on each valid data transfer. If the counter expires before the next assertion of TRDY# is received, the Core Logic module stops the transaction with a master abort and asserts SERR#, if enabled to do so (via F0 Index 04h[8]).
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 80h**
7:0	**PCI Header Type Register.** This register defines the format of this header. This header has a format of type 0. (For more information about this format, see the PCI Local Bus specification, revision 2.2.) Additionally, bit 7 of this register defines whether this PCI device is a multifunction device (bit 7 = 1) or not (bit 7 = 0).
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
This register indicates various information about the PCI Built-In Self-Test (BIST) mechanism. **Note:** This mechanism is not supported in the Core Logic module in the SC2200.	
7	**BIST Capable.** Indicates if the device can run a Built-In Self-Test (BIST). 0: The device has no BIST functionality. 1: The device can run a BIST.
6	**Start BIST.** Setting this bit to 1 starts up a BIST on the device. The device resets this bit when the BIST is completed. (Not supported.)
5:4	**Reserved.**
3:0	**BIST Completion Code.** Upon completion of the BIST, the completion code is stored in these bits. A completion code of 0000 indicates that the BIST was successfully completed. Any other value indicates a BIST failure.
Index 10h-13h	**Base Address Register 0 - F0BAR0 (R/W)** — **Reset Value: 00000001h**
This register allows access to I/O mapped GPIO runtime and configuration Registers. Bits [5:0] are read only (000001), indicating a 64-byte aligned I/O address space. Refer to Table 5-30 on page 231 for the GPIO register bit formats and reset values.	
31:6	**GPIO Base Address.**
5:0	**Address Range. (Read Only)**
Index 14h-17h	**Base Address Register 1 - F0BAR1 (R/W)** — **Reset Value: 00000001h**
This register allows access to I/O mapped LPC configuration registers. Bits [5:0] are read only (000001), indicating a 64-byte aligned I/O address space. Refer to Table 5-31 on page 235 for the bit formats and reset values of the LPC registers.	
31:6	**LPC Base Address.**
5:0	**Address Range. (Read Only)**
Index 18h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0500h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 40h	**PCI Function Control Register 1 (R/W)** — **Reset Value: 39h**
7:6	**Reserved.** Must be set to 0.
5	**Reserved.** Must be set to 0.
4	**PCI Subtractive Decode.** 0: Disable transfer of subtractive decode address to external PCI bus. External PCI bus is not usable. 1: Enable transfer of subtractive decode address to external PCI bus. Recommended setting.
3	**Reserved**. Must be set to 1.
2	**Reserved**. Must be set to 0.
1	**PERR# Signals SERR#.** Assert SERR# when PERR# is asserted or detected as active by the Core Logic module (allows PERR# assertion to be cascaded to NMI (SMI) generation in the system). 0: Disable. 1: Enable.
0	**PCI Interrupt Acknowledge Cycle Response.** The Core Logic module responds to PCI interrupt acknowledge cycles. 0: Disable. 1: Enable.
Index 41h	**PCI Function Control Register 2 (R/W)** — **Reset Value: 00h**
7:6	**Reserved.** Must be set to 0.
5	**X-Bus Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 5 (F5) register space, an SMI is generated. Writes are trapped; access to the register is denied. Reads are snooped; access to the register is allowed. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
4	**Video Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 4 (F4) register space, an SMI is generated.Writes are trapped; access to the register is denied. Reads are snooped; access to the register is allowed. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
3	**Audio Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 3 (F3) register space, an SMI is generated. Writes are trapped; access to the register is denied. Reads are snooped; access to the register is allowed. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
2	**IDE Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 2 (F2) register space, an SMI is generated. Writes are trapped; access to the register is denied. Reads are snooped; access to the register is allowed. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
1	**Power Management Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 1 (F1) register space, an SMI is generated. Writes are trapped; access to the register is denied. Reads are snooped; access to the register is allowed. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
0	**Legacy Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 0 (F0), an SMI is generated. Reads and writes are snooped; access to the register is allowed. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
Index 42h	**Reserved** — **Reset Value: 00h**
Index 43h	**Delayed Transactions Register (R/W)** — **Reset Value: 02h**
7:6	**Reserved.** Must be set to 0.
5	**Reserved.** Must be set to 1.
4	**Enable PCI Delayed Transactions for Access to I/O Address 170h-177h (Secondary IDE Channel).** PIO mode uses repeated I/O transactions that are faster when non-delayed transactions are used. 0: I/O addresses complete as fast as possible on PCI. (Default) 1: Accesses to Secondary IDE channel I/O addresses are delayed transactions on PCI. For best performance of VIP, this bit should be set to 1 unless PIO mode 3 or 4 are used.
3	**Enable PCI Delayed Transactions for Access to I/O Address 1F0h-1F7h (Primary IDE Channel).** PIO mode uses repeated I/O transactions that are faster when non-delayed transactions are used. 0: I/O addresses complete as fast as possible on PCI. (Default) 1: Accesses to Primary IDE channel I/O addresses are delayed transactions on PCI. For best performance of VIP, this bit should be set to 1 unless PIO mode 3 or 4 are used.
2	**Enable PCI Delayed Transactions for AT Legacy PIC I/O Addresses.** Some PIC status reads are long. Enabling delayed transactions help reduce DMA latency for high bandwidth devices like VIP. 0: PIC I/O addresses complete as fast as possible on PCI. (Default) 1: Accesses to PIC I/O addresses are delayed transactions on PCI. For best performance of VIP, this bit should be set to 1.
1	**Enable PCI Delayed Transactions for AT Legacy PIT I/O Addresses.** Some x86 programs (certain benchmarks/diagnostics) assume a particular latency for PIT accesses; this bit allows that code to work. 0: PIT I/O addresses complete as fast as possible on PCI. 1: Accesses to PIT I/O addresses are delayed transactions on PCI. (Default) For best performance (e.g., when running Microsoft Windows), this bit should be set to 0.
0	**Reserved.** Must be set to 0.
Index 44h	**Reset Control Register (R/W)** — **Reset Value: 01h**
7	**AC97 Soft Reset.** Active low reset for the AC97 codec interface. 0: AC97_RST# is driven high. (Default) 1: AC97_RST# is driven low.
6:4	**Reserved.** Must be set to 0.
3	**IDE Controller Reset.** Reset the IDE controller. 0: Disable. 1: Enable. Write 0 to clear. This bit is level-sensitive and must be cleared after the reset is enabled.
2	**IDE Reset.** Reset IDE bus. 0: Disable. 1: Enable (drives outputs to zero). Write 0 to clear. This bit is level-sensitive and must be cleared after the reset is enabled.
1	**PCI Reset.** Reset PCI bus. 0: Disable. 1: Enable. When this bit is set to 1, the Core Logic module output signal PCIRST# is asserted and all devices on the PCI bus (including PCIUSB) are reset. No other function within the Core Logic module is affected by this bit. Write 0 to clear this bit. This bit is level-sensitive and must be cleared after the reset is enabled.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
0	**X-Bus Warm Start.** Writing and reading this bit each have different meanings. When reading this bit, it indicates whether or not a warm start occurred since power-up: 0: A warm start occurred. 1: No warm start has occurred. When writing this bit, it can be used to trigger a system-wide reset: 0: No effect. 1: Execute system-wide reset (used only for clock configuration at power-up).
Index 45h	**Reserved** — **Reset Value: 00h**
Index 46h	**PCI Functions Enable Register (R/W)** — **Reset Value: FEh**
7:6	**Reserved.** Resets to 11.
5	**F5 (PCI Function 5).** When asserted (set to 1), enables the register space designated as F5. This bit must always be set to 1. (Default)
4	**F4 (PCI Function 4).** When asserted (set to 1), enables the register space designated as F4. This bit must always be set to 1. (Default)
3	**F3 (PCI Function 3).** When asserted (set to 1), enables the register space designated as F3. This bit must always be set to 1. (Default)
2	**F2 (PCI Function 2).** When asserted (set to 1), enables the register space designated as F2. This bit must always be set to 1. (Default)
1	**F1 (PCI Function 1).** When asserted (set to 1), enables the register space designated as F1. This bit must always be set to 1. (Default)
0	**Reserved.** Must be set to 0.
Index 47h	**Miscellaneous Enable Register (R/W)** — **Reset Value: 00h**
7:3	**Reserved.** Must be set to 0.
2	**F0BAR1 (PCI Function 0, Base Address Register 1).** F0BAR1, pointer to I/O mapped LPC configuration registers. 0: Disable. 1: Enable.
1	**F0BAR0 (PCI Function 0, Base Address Register 0).** F0BAR0, pointer to I/O mapped GPIO configuration registers. 0: Disable. 1: Enable.
0	**Reserved.** Must be set to 0.
Index 48h-4Bh	**Reserved** — **Reset Value: 00h**
Index 4Ch-4Fh	**Top of System Memory (R/W)** — **Reset Value: FFFFFFFFh**
31:0	**Top of System Memory.** Highest address in system used to determine active decode for external PCI mastered memory cycles. If an external PCI master requests a memory address below the value programmed in this register, the cycle is transferred from the external PCI bus interface to the Fast-PCI interface for servicing by the GX1 module. **Note:** The four least significant bits must be set to 1100.
Index 50h	**PIT Control/ISA CLK Divider (R/W)** — **Reset Value: 7Bh**
7	**PIT Software Reset.** 0: Disable. 1: Enable.
6	**PIT Counter 1.** 0: Forces Counter 1 output (OUT1) to zero. 1: Allows Counter 1 output (OUT1) to pass to the Port 061h[4].
5	**PIT Counter 1 Enable.** 0: Sets GATE1 input low. 1: Sets GATE1 input high.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4	**PIT Counter 0.** 0: Forces Counter 0 output (OUT0) to zero. 1: Allows Counter 0 output (OUT0) to pass to IRQ0.
3	**PIT Counter 0 Enable.** 0: Sets GATE0 input low. 1: Sets GATE0 input high.
2:0	**ISA Clock Divisor.** Determines the divisor of the PCI clock used to make the ISA clock, which is typically programmed for approximately 8 MHz: 000: Divide by 1 — 100: Divide by 5 001: Divide by 2 — 101: Divide by 6 010: Divide by 3 — 110: Divide by 7 011: Divide by 4 — 111: Divide by 8 If PCI clock = 25 MHz, use setting of 010 (divide by 3). If PCI clock = 30 or 33 MHz, use a setting of 011 (divide by 4).
Index 51h	**ISA I/O Recovery Control Register (R/W)** — **Reset Value: 40h**
7:4	**8-Bit I/O Recovery.** These bits determine the number of ISA bus clocks between back-to-back 8-bit I/O read cycles. This count is in addition to a preset one-clock delay built into the controller. 0000: 1 PCI clock 0001: 2 PCI clocks ::: ::: ::: 1111: 16 PCI clocks
3:0	**16-Bit I/O Recovery.** These bits determine the number of ISA bus clocks between back-to-back 16-bit I/O cycles. This count is in addition to a preset one-clock delay built into the controller. 0000: 1 PCI clock 0001: 2 PCI clocks ::: ::: ::: 1111: 16 PCI clocks
Index 52h	**ROM/AT Logic Control Register (R/W)** — **Reset Value: 98h**
7	**Snoop Fast Keyboard Gate A20 and Fast Reset.** Enables the snoop logic associated with keyboard commands for A20 Mask and Reset. 0: Disable snooping. The keyboard controller handles the commands. 1: Enable snooping.
6:5	**Reserved.** Must be set to 0.
4	**Enable A20M# De-assertion on Warm Reset.** Force A20M# high during a Warm Reset (guarantees that A20M# is de-asserted regardless of the state of A20). 0: Disable. 1: Enable.
3	**Enable Port 092h (Port A).** Port 092h decode and the logical functions. 0: Disable. 1: Enable.
2	**Upper ROM Size.** Selects upper ROM addressing size. 0: 256K (FFFC0000h-FFFFFFFFh). 1: Use ROM Mask register (F0 Index 6Eh). ROMCS# goes active for the above ranges whether strapped for ISA or LPC. (Refer to F0BAR1+I/O Offset 10h[15] for further strapping/programming details.) The selected range can then be either positively or subtractively decoded through F0 Index 5Bh[5].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**ROM Write Enable**. When asserted, enables writes to ROM space, allowing Flash programming. If strapped for ISA and this bit is set to 1, writes to the configured ROM space asserts ROMCS#, enabling the write cycle to the Flash device on the ISA bus. Otherwise, ROMCS# is inhibited for writes. If strapped for LPC and this bit is set to 1, the cycle runs on the LPC bus. Otherwise, the LPC bus cycle is inhibited for writes. Refer to F0BAR1+I/O Offset 10h[15] for further strapping/programming details.
0	**Lower ROM Size.** Selects lower ROM addressing size in which ROMCS# goes active. 0: Lower ROM access are 000F0000h-000FFFFFh (64 KB). (Default) 1: Lower ROM accesses are 000E0000h-000FFFFFh (128 KB). ROMCS# goes active for the above ranges whether strapped for ISA or LPC. (Refer to F0BAR1+I/O Offset 10h[15] for further strapping/programming details.) The selected range can then be either positively or subtractively decoded through F0 Index 5Bh[5].

Index 53h	Alternate CPU Support Register (R/W)	Reset Value: 00h
7:6	**Reserved.** Must be set to 0.	
5	**Bidirectional SMI Enable.** 0: Disable. 1: Enable. This bit must be set to 0.	
4:3	**Reserved.** Must be set to 0.	
2	**Reserved.** Must be set to 0.	
1	**IRQ13 Function Selection.** Selects function of the internal IRQ13/FERR# signal. 0: FERR#. 1: IRQ13. This bit must be set to 1.	
0	**Generate SMI on A20M# Toggle.** 0: Disable. 1: Enable. This bit must be set to 1. SMI status is reported at F1BAR0+I/O Offset 00h/02h[7].	

Index 54h-59h	Reserved	Reset Value: 00h

Index 5Ah	Decode Control Register 1 (R/W)	Reset Value: 01h
Indicates PCI positive or negative decoding for various I/O ports on the ISA bus. **Note:** Positive decoding by the Core Logic module speeds up I/O cycle time. The I/O ports mentioned in the bit descriptions below, do not exist in the Core Logic module. It is assumed that if positive decode is enabled for a port, the port exists on the ISA bus.		
7	**Secondary Floppy Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 372h-375h and 377h. 0: Subtractive. 1: Positive.	
6	**Primary Floppy Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3F2h-3F5h and 3F7h. 0: Subtractive. 1: Positive.	
5	**COM4 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 2E8h-2EFh. 0: Subtractive. 1: Positive.	
4	**COM3 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3E8h-3EFh. 0: Subtractive. 1: Positive.	

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
3	**COM2 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 2F8h-2FFh. 0: Subtractive. 1: Positive.
2	**COM1 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3F8h-3FFh. 0: Subtractive. 1: Positive.
1	**Keyboard Controller Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O Ports 060h and 064h (as well as 062h and 066h, if enabled - F4 Index 5Bh[7] = 1). 0: Subtractive. 1: Positive. **Note:** If F0BAR1+I/O Offset 10h bits 10 = 0 and 16 = 1, then this bit must be written 0.
0	**Real-Time Clock Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O Ports 070h-073h. 0: Subtractive. 1: Positive.
Index 5Bh	**Decode Control Register 2 (R/W)** **Reset Value: 20h**
Note:	Positive decoding by the Core Logic module speeds up the I/O cycle time. The Keyboard, LPT3, LPT2, and LPT1 I/O ports do not exist in the Core Logic module. It is assumed that if positive decoding is enabled for any of these ports, the port exists on the ISA bus.
7	**Keyboard I/O Port 062h/066h Positive Decode.** This alternate port to the keyboard controller is provided in support of power management features. 0: Disable. 1: Enable.
6	**Reserved.** Must be set to 0.
5	**BIOS ROM Positive Decode.** Selects PCI positive or subtractive decoding for accesses to the configured ROM space. 0: Subtractive. 1: Positive. ROM configuration is at F0 Index 52h[2:0].
4	**Secondary IDE Controller Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 170h-177h and 376h-377h (excluding writes to 377h). 0: Subtractive. Subtractively decoded IDE addresses are forwarded to the PCI slot bus. If a master abort occurs, they are then forwarded to ISA. 1: Positive. Positively decoded IDE addresses are forwarded to the internal IDE controller and then to the IDE bus.
3	**Primary IDE Controller Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 1F0h-1F7h and 3F6h-3F7h (excluding writes to 3F7h). 0: Subtractive. Subtractively decoded IDE addresses are forwarded to the PCI slot bus. If a master abort occurs, they are then forwarded to ISA. 1: Positive. Positively decoded IDE addresses are forwarded to the internal IDE controller and then to the IDE bus.
2	**LPT3 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 278h-27Fh. 0: Subtractive. 1: Positive.
1	**LPT2 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 378h-37Fh. 0: Subtractive. 1: Positive.
0	**LPT1 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3BCh-3BFh 0: Subtractive. 1: Positive.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 5Ch	**PCI Interrupt Steering Register 1 (R/W)** — **Reset Value: 00h**
	Indicates target interrupts for signals INTB# and INTA#.
	Note: The target interrupt must first be configured as level sensitive via I/O Ports 4D0h and 4D1h in order to maintain PCI interrupt compatibility.
7:4	**INTB# (EBGA Ball AF1 / TEPBGA Ball C26) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15
3:0	**INTA# (EBGA Ball AE3 / TEPBGA Ball D26) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15
Index 5Dh	**PCI Interrupt Steering Register 2 (R/W)** — **Reset Value: 00h**
	Indicates target interrupts for signals INTD# and INTC#. Note that INTD# is muxed with IDE_DATA7 (selection made via PMR[24]) and INTC# is muxed with GPIO19+IOCHRDY (selection made via PMR[9,4]). See Table 3-2 on page 81 for PMR bit descriptions.
	Note: The target interrupt must first be configured as level sensitive via I/O Ports 4D0h and 4D1h in order to maintain PCI interrupt compatibility.
7:4	**INTD# (EBGA Ball B22 / TEPBGA Ball AA2) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15
3:0	**INTC# (EBGA Ball H4 / TEPBGA Ball C9) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15
Index 5Eh-5Fh	**Reserved** — **Reset Value: 00h**
Index 60h-63h	**ACPI Control Register (R/W)** — **Reset Value: 00000000h**
31:8	**Reserved.** Must be set to 0.
7	**SUSP_3V Shut Down PLL5.** Allow internal SUSP_3V to shut down PLL5. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
6	**SUSP_3V Shut Down PLL4.** Allow internal SUSP_3V to shut down PLL4 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
5	**SUSP_3V Shut Down PLL3.** Allow internal SUSP_3V to shut down PLL3. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active..
4	**SUSP_3V Shut Down PLL2.** Allow internal SUSP_3V to shut down PLL2. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
3	**SUSP_3V Shut Down PLL6.** Allow internal SUSP_3V to shut down PLL6. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
2	**ACPI C3 SUSP_3V Enable.** Allow internal SUSP_3V to be active during C3 state. 0: Disable. 1: Enable.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**ACPI SL1 SUSP_3V Enable.** Allow internal SUSP_3V to be active during SL1 sleep state. 0: Disable. 1: Enable.
0	**ACPI C3 Support Enable.** Allow support of C3 states. 0: Disable. 1: Enable.
Index 64h-6Dh	**Reserved** — **Reset Value: 00h**
Index 6Eh-6Fh	**ROM Mask Register (R/W)** — **Reset Value: FFF0h**
15:8	**Reserved.** Must be set to FFh.
7:4	**ROM Size.** If F0 Index 52h[2] = 1: 0000: 16 MB = FF000000h-FFFFFFFFh 1000: 8 MB = FF800000h-FFFFFFFFh 1100: 4 MB = FFC00000h-FFFFFFFFh 1110: 2 MB = FFE00000h-FFFFFFFFh 1111: 1 MB = FFF00000h-FFFFFFFFh All other settings for these bits are reserved.
3:0	**Reserved.** Must be set to 0.
Index 70h-71h	**IOCS1# Base Address Register (R/W)** — **Reset Value: 0000h**
15:0	**I/O Chip Select 1 Base Address.** This 16-bit value represents the I/O base address used to enable assertion of IOCS1# (EBGA ball H2 or AL12 / TEPBGA ball D10 or N30 - see PMR[23] in Table 3-2 on page 81). This register is used in conjunction with F0 Index 72h (IOCS1# Control register).
Index 72h	**IOCS1# Control Register (R/W)** — **Reset Value: 00h** This register is used in conjunction with F0 Index 70h (IOCS1# Base Address register).
7	**I/O Chip Select 1 Positive Decode (IOCS1#).** 0: Disable. 1: Enable.
6	**Writes Result in Chip Select.** When this bit is set to 1, writes to configured I/O address (base address configured in F0 Index 70h; range configured in bits [4:0]) cause IOCS1# to be asserted. 0: Disable. 1: Enable.
5	**Reads Result in Chip Select.** When this bit is set to 1, reads from configured I/O address (base address configured in F0 Index 70h; range configured in bits [4:0]) cause IOCS1# to be asserted. 0: Disable. 1: Enable.
4:0	**IOCS1# I/O Address Range.** This 5-bit field is used to select the range of IOCS1#. 00000: 1 Byte 00001: 2 Bytes 00011: 4 Bytes 00111: 8 Bytes 01111: 16 Bytes 11111: 32 Bytes All other combinations are reserved.
Index 73h	**Reserved** — **Reset Value: 00h**
Index 74h-75h	**IOCS0# Base Address Register (R/W)** — **Reset Value: 0000h**
15:0	**I/O Chip Select 0 Base Address.** This 16-bit value represents the I/O base address used to enable the assertion of IOCS0# (EBGA ball J4 / TEPBGA ball A10 - see PMR[23] in Table 3-2 on page 81). This register is used in conjunction with F0 Index 76h (IOCS0# Control register).

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 76h	**IOCS0# Control Register (R/W)** — **Reset Value: 00h**
This register is used in conjunction with F0 Index 74h (IOCS0# Base Address register).	
7	**I/O Chip Select 0 Positive Decode (IOCS0#).** 0: Disable. 1: Enable.
6	**Writes Result in Chip Select.** When this bit is set to 1, writes to configured I/O address (base address configured in F0 Index 74h; range configured in bits [4:0]) cause IOCS0# to be asserted. 0: Disable. 1: Enable.
5	**Reads Result in Chip Select.** When this bit is set to 1, reads from configured I/O address (base address configured in F0 Index 74h; range configured in bits [4:0]) cause IOCS0# to be asserted. 0: Disable. 1: Enable.
4:0	**IOCS0# I/O Address Range.** This 5-bit field is used to select the range of IOCS0#. 00000: 1 Byte 00001: 2 Bytes 00011: 4 Bytes 00111: 8 Bytes 01111: 16 Bytes 11111: 32 Bytes All other combinations are reserved.
Index 77h	**Reserved** — **Reset Value: 00h**
Index 78h-7Bh	**DOCCS# Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**DiskOnChip Chip Select Base Address.** This 32-bit value represents the memory base address used to enable assertion of DOCCS# (EBGA ball H3 or AJ13 / TEPBGA ball A9 or N31, see PMR[23] in Table 3-2 on page 81). This register is used in conjunction with F0 Index 7Ch (DOCCS# Control register).
Index 7Ch-7Fh	**DOCCS# Control Register (R/W)** — **Reset Value: 00000000h**
This register is used in conjunction with F0 Index 78h (DOCCS# Base Address register).	
31:27	**Reserved.** Must be set to 0.
26	**DiskOnChip Chip Select Positive Decode (DOCCS#).** 0: Disable. 1: Enable.
25	**Writes Result in Chip Select.** When this bit is set to 1, writes to configured memory address (base address configured in F0 Index 78h; range configured in bits [18:0]) cause DOCCS# to be asserted. 0: Disable. 1: Enable.
24	**Reads Result in Chip Select.** When this bit is set to 1, reads from configured memory address (base address configured in F0 Index 78h; range configured in bits [18:0]) cause DOCCS# to be asserted. 0: Disable. 1: Enable.
23:19	**Reserved.** Must be set to 0.
18:0	**DOCCS# Memory Address Range.** This 19-bit mask is used to qualify accesses on which DOCCS# is asserted by masking the upper 19 bits of the incoming PCI address (AD[31:13]).
Index 80h	**Power Management Enable Register 1 (R/W)** — **Reset Value: 00h**
7:6	**Reserved.** Must be set to 0.
5	**Codec SDATA_IN SMI.** When set to 1, this bit allows an SMI to be generated in response to an AC97 codec producing a positive edge on SDATA_IN. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 87h/F7h[2].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4	**Video Speedup.** Any video activity, as decoded from the serial connection (PSERIAL) from the GX1 module disables clock throttling (via internal SUSP#/SUSPA# handshake) for a configurable duration when system is power-managed using CPU Suspend modulation. 0: Disable. 1: Enable. The duration of the speedup is configured in the Video Speedup Timer Count Register (F0 Index 8Dh). Detection of an external VGA access (3Bx, 3Cx, 3Dx and A000h-B7FFh) on the PCI bus is also supported. This configuration is non-standard, but it does allow the power management routines to support an external VGA chip.
3	**IRQ Speedup.** Any unmasked IRQ (per I/O Ports 021h/0A1h) or SMI disables clock throttling (via internal SUSP#/SUSPA# handshake) for a configurable duration when system is power-managed using CPU Suspend modulation. 0: Disable. 1: Enable. The duration of the speedup is configured in the IRQ Speedup Timer Count Register (F0 Index 8Ch).
2	**Traps.** Globally enable all power management I/O traps. 0: Disable. 1: Enable. This excludes the audio I/O traps, which are enabled via F3BAR0+Memory Offset 18h.
1	**Idle Timers.** Device idle timers. 0: Disable. 1: Enable. **Note:** Disable at this level does not reload the timers on the enable. The timers are disabled at their current counts. This bit has no affect on the Suspend Modulation register (F0 Index 94h). Only applicable when in APM mode (F1BAR1+I/O Offset 0Ch[0] = 0) and not ACPI mode.
0	**Power Management.** Global power management. 0: Disable. 1: Enable. This bit must be set to 1 immediately after POST for power management resources to function.
Index 81h	**Power Management Enable Register 2 (R/W)** — **Reset Value: 00h**
7	**Video Access Idle Timer Enable.** Turn on Video Idle Timer Count Register (F0 Index A6h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the video address range (sets bit 0 of the GX1 module's PSERIAL register) the timer is reloaded with the programmed count. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[7].
6	**User Defined Device 3 (UDEF3) Idle Timer Enable.** Turn on UDEF3 Idle Timer Count Register (F0 Index A4h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. UDEF3 address programming is at F0 Index C8h (base address register) and CEh (control register). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[6].
5	**User Defined Device 2 (UDEF2) Idle Timer Enable.** Turn on UDEF2 Idle Timer Count Register (F0 Index A2h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. UDEF2 address programming is at F0 Index C4h (base address register) and CDh (control register). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[5].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4	**User Defined Device 1 (UDEF1) Idle Timer Enable.** Turn on UDEF1 Idle Timer Count Register (F0 Index A0h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. UDEF1 address programming is at F0 Index C0h (base address register) and CCh (control register). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[4].
3	**Keyboard/Mouse Idle Timer Enable.** Turn on Keyboard/Mouse Idle Timer Count Register (F0 Index 9Eh) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges listed below, the timer is reloaded with the programmed count: — Keyboard Controller: I/O Ports 060h/064h. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is included). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is included). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[3].
2	**Parallel/Serial Idle Timer Enable.** Turn on Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges listed below, the timer is reloaded with the programmed count. — LPT1: I/O Port 3BCh-3BEh. — LPT2: I/O Port 378h-37Fh. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is excluded). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is excluded). — COM3: I/O Port 3E8h-3EFh. — COM4: I/O Port 2E8h-2EFh. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[2].
1	**Floppy Disk Idle Timer Enable.** Turn on Floppy Disk Idle Timer Count Register (F0 Index 9Ah) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges (listed below, the timer is reloaded with the programmed count. — Primary floppy disk: I/O Port 3F2h-3F5h, 3F7h. — Secondary floppy disk: I/O Port 372h-375h, 377h. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[1].
0	**Primary Hard Disk Idle Timer Enable.** Turn on Primary Hard Disk Idle Timer Count Register (F0 Index 98h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges selected in F0 Index 93h[5], the timer is reloaded with the programmed count. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[0].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 82h	**Power Management Enable Register 3 (R/W)** **Reset Value: 00h**
7	**Video Access Trap.** If this bit is enabled and an access occurs in the video address range (sets bit 0 of the GX1 module's PSERIAL register), an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[7].
6	**User Defined Device 3 (UDEF3) Access Trap.** If this bit is enabled and an access occurs in the programmed address range, an SMI is generated. UDEF3 address programming is at F0 Index C8h (Base Address register) and CEh (Control register). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[4].
5	**User Defined Device 2 (UDEF2) Access Trap.** If this bit is enabled and an access occurs in the programmed address range, an SMI is generated. UDEF2 address programming is at F0 Index C4h (Base Address register) and CDh (Control register). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[3].
4	**User Defined Device 1 (UDEF1) Access Trap.** If this bit is enabled and an access occurs in the programmed address range, an SMI is generated. UDEF1 address programming is at F0 Index C0h (base address register), and CCh (control register). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[2].
3	**Keyboard/Mouse Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges listed below, an SMI is generated. — Keyboard Controller: I/O Ports 060h/064h. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is included). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is included). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[3].
2	**Parallel/Serial Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges listed below, an SMI is generated. — LPT1: I/O Port 3BCh-3BEh. — LPT2: I/O Port 378h-37Fh. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is excluded). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is excluded). — COM3: I/O Port 3E8h-3EFh. — COM4: I/O Port 2E8h-2EFh. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[2].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**Floppy Disk Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges listed below, an SMI is generated. — Primary floppy disk: I/O Port 3F2h-3F5h, 3F7h. — Secondary floppy disk: I/O Port 372h-375h, 377h. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[1].
0	**Primary Hard Disk Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges selected in F0 Index 93h[5], an SMI is generated. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[0].
Index 83h	**Power Management Enable Register 4 (R/W)** — **Reset Value: 00h**
7	**Secondary Hard Disk Idle Timer Enable.** Turn on Secondary Hard Disk Idle Timer Count Register (F0 Index ACh) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges selected in F0 Index 93h[4], the timer is reloaded with the programmed count. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[4].
6	**Secondary Hard Disk Access Trap.** If this bit is enabled and an access occurs in the address ranges selected in F0 Index 93h[4], an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[5].
5	**ACPI Timer SMI.** Allow SMI generation for MSB toggles on the ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 87h/F7h[0].
4	**THRM# SMI.** Allow SMI generation on assertion of THRM#. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 87h/F7h[6].
3	**VGA Timer Enable.** Turn on VGA Timer Count Register (F0 Index 8Eh) and generate an SMI when the timer reaches 0. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. F0 Index 8Bh[6] selects the timebase for the VGA Timer. SMI status is reported at F1BAR0+I/O Offset 00h/02h[6] (top level only).
2	**Video Retrace Interrupt SMI.** Allow SMI generation whenever video retrace occurs. 0: Disable. 1: Enable. This information is decoded from the serial connection (PSERIAL register, bit 7) from the GX1 module. This function is normally not used for power management but for soft (VSA) VGA routines. SMI status reporting is at F1BAR0+I/O Offset 00h/02h[5] (top level only).

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**General Purpose Timer 2 Enable.** Turn on GP Timer 2 Count Register (F0 Index 8Ah) and generate an SMI when the timer expires. 0: Disable. 1: Enable. This idle timer is reloaded from the assertion of GPIO7 (if programmed to do so). GP Timer 2 programming is at F0 Index 8Bh[5,3,2]. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[1].
0	**General Purpose Timer 1 Enable.** Turn on GP Timer 1 Count Register (F0 Index 88h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. This idle timer's load is multi-sourced and gets reloaded any time an enabled event (F0 Index 89h[6:0]) occurs. GP Timer 1 programming is at F0 Index 8Bh[4]. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[0].

Index 84h — Second Level PME/SMI Status Mirror Register 1 (RO) — Reset Value: 00h

The bits in this register are used for the second level of status reporting. The top level is reported at F1BAR0+I/O Offset 00h/02h[0].

This register is called a "mirror" register since an identical register exists at F0 Index F4h. Reading this register does not clear the status, while reading its counterpart at F0 Index F4h clears the status at both the second and the top levels.

7:3	**Reserved.** Reads as 0.
2	**GPWIO2 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO2 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO2 is enabled as an input: F1BAR1+I/O Offset 15h[2] = 0. 2) Set F1BAR1+I/O Offset 15h[6] to 1.
1	**GPWIO1 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO1 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO1 is enabled as an input: F1BAR1+I/O Offset 15h[1] = 0. 2) Set F1BAR1+I/O Offset 15h[5] to 1.
0	**GPWIO0 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO0 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO0 is enabled as an input: F1BAR1+I/O Offset 15h[0] = 0. 2) Set F1BAR1+I/O Offset 15h[4] to 1.

Index 85h — Second Level PME/SMI Status Mirror Register 2 (RO) — Reset Value: 00h

The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0].

This register is called a “Mirror” register since an identical register exists at F0 Index F5h. Reading this register does not clear the status, while reading its counterpart at F0 Index F5h clears the status at both the second and top levels.

7	**Video Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Video Idle Timer Count Register (F0 Index A6h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[7] to 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
6	**User Defined Device Idle Timer 3 Timeout.** Indicates whether or not an SMI was caused by expiration of User Defined Device 3 Idle Timer Count Register (F0 Index A4h). 0: No 1: Yes To enable SMI generation, set F0 Index 81h[6] to 1.
5	**User Defined Device Idle Timer 2 Timeout.** Indicates whether or not an SMI was caused by expiration of User Defined Device 2 Idle Timer Count Register (F0 Index A2h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[5] to 1.
4	**User Defined Device Idle Timer 1 Timeout.** Indicates whether or not an SMI was caused by expiration of User Defined Device 1 Idle Timer Count Register (F0 Index A0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[4] to 1.
3	**Keyboard/Mouse Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Keyboard/Mouse Idle Timer Count Register (F0 Index 9Eh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[3] to 1.
2	**Parallel/Serial Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[2] to 1.
1	**Floppy Disk Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Floppy Disk Idle Timer Count Register (F0 Index 9Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[1] to 1.
0	**Primary Hard Disk Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Primary Hard Disk Idle Timer Count Register (F0 Index 98h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[0] to 1.

Index 86h — Second Level PME/SMI Status Mirror Register 3 (RO) — Reset Value: 00h

The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0].

This register is called a “Mirror” register since an identical register exists at F0 Index F6h. Reading this register does not clear the status, while reading its counterpart at F0 Index F6h clears the status at both the second and top levels.

7	**Video Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the Video I/O Trap. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[7] to 1.
6	**Reserved**
5	**Secondary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the secondary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[6] to 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4	**Secondary Hard Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Secondary Hard Disk Idle Timer Count register (F0 Index ACh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[7] to 1.
3	**Keyboard/Mouse Access Trap SMI Status.** Indicates whether or not an SMI was caused by an trapped I/O access to the keyboard or mouse. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[3] to 1.
2	**Parallel/Serial Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to either the serial or parallel ports. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[2] to 1.
1	**Floppy Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the floppy disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[1] to 1.
0	**Primary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the primary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[0] to 1.
Index 87h	**Second Level PME/SMI Status Mirror Register 4 (RO)** **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported at F1BAR0+I/O Offset 00h/02h[0]. This register is called a “Mirror” register since an identical register exists at F0 Index F7h. Reading this register does not clear the status, while reading its counterpart at F0 Index F7h clears the status at both the second and top levels except for bit 7 which has a third level of SMI status reporting at F0BAR0+I/O 0Ch/1Ch.	
7	**GPIO Event SMI Status.** Indicates whether or not an SMI was caused by a transition of any of the GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] to 0. F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled to generate a PME and setting F1BAR1+I/O Offset 0Ch[0] = 0 enables the PME to generate an SMI. In addition, the selected GPIO must be enabled as an input (F0BAR0+I/O Offset 20h and 24h). The next level (third level) of SMI status is at F0BAR0+I/O 0Ch/1Ch[15:0].
6	**Thermal Override SMI Status.** Indicates whether or not an SMI was caused by the assertion of THRM#. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[4] to 1.
5:4	**Reserved.** Always reads 0.
3	**SIO PWUREQ SMI Status.** Indicates whether or not an SMI was caused by a power-up event from the SIO. 0: No. 1: Yes. A power-up event is defined as any of the following events/activities: — RI2# — SDATA_IN2 — IRRX1 (CEIR) To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] to 0.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Codec SDATA_IN SMI Status.** Indicates whether or not an SMI was caused by AC97 Codec producing a positive edge on SDATA_IN. 0: No. 1: Yes. To enable SMI generation, set F0 Index 80h[5] to 1.
1	**RTC Alarm (IRQ8#) SMI Status.** Indicates whether or not an SMI was caused by an RTC interrupt. 0: No. 1: Yes. This SMI event can only occur while in 3V Suspend and an RTC interrupt occurs with F1BAR1+I/O Offset 0Ch[0] = 0.
0	**ACPI Timer SMI Status.** Indicates whether or not an SMI was caused by an ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch) MSB toggle. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[5] to 1.
Index 88h	**General Purpose Timer 1 Count Register (R/W)** **Reset Value: 00h**
7:0	**GPT1_COUNT.** This field represents the load value for General Purpose Timer 1. This value can represent either an 8-bit counter or a 16-bit counter (selected in F0 Index 8Bh[4]). It is loaded into the counter when the timer is enabled (F0 Index 83h[0] = 1). Once enabled, an enabled event (configured in F0 Index 89h[6:0]) reloads the timer. The counter is decremented with each clock of the configured timebase (1 msec or 1 sec selected at F0 Index 89h[7]). Upon expiration of the counter, an SMI is generated, and the top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. The second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[0]. Once expired, this counter must be re-initialized by either disabling and enabling it, or writing a new count value in this register.
Index 89h	**General Purpose Timer 1 Control Register (R/W)** **Reset Value: 00h**
7	**General Purpose Timer 1 TImebase.** Selects timebase for General Purpose Timer 1 (F0 Index 88h). 0: 1 second. 1: 1 millisecond.
6	**Re-trigger General Purpose Timer 1 on User Defined Device 3 (UDEF3) Activity.** 0: Disable. 1: Enable. Any access to the configured (memory or I/O) address range for UDEF3 (configured in F0 Index C8h and CEh) reloads General Purpose Timer 1.
5	**Re-trigger General Purpose Timer 1 on User Defined Device 2 (UDEF2) Activity.** 0: Disable. 1: Enable. Any access to the configured (memory or I/O) address range for UDEF2 (configured in F0 Index C4h and CDh) reloads General Purpose Timer 1.
4	**Re-trigger General Purpose Timer 1 on User Defined Device 1 (UDEF1) Activity**. 0: Disable. 1: Enable. Any access to the configured (memory or I/O) address range for UDEF1 (configured in F0 Index C0h and CCh) reloads General Purpose Timer 1.
3	**Re-trigger General Purpose Timer 1 on Keyboard or Mouse Activity.** 0: Disable. 1: Enable. Any access to the keyboard or mouse I/O address range listed below reloads General Purpose Timer 1: — Keyboard Controller: I/O Ports 060h/064h. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is included). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is included).

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Re-trigger General Purpose Timer 1 on Parallel/Serial Port Activity.** 0: Disable. 1: Enable. Any access to the parallel or serial port I/O address range listed below reloads the General Purpose Timer 1: — LPT1: I/O Port 3BCh-3BEh. — LPT2: I/O Port 378h-37Fh. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is excluded). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is excluded). — COM3: I/O Port 3E8h-3EFh. — COM4: I/O Port 2E8h-2EFh.
1	**Re-trigger General Purpose Timer 1 on Floppy Disk Activity.** 0: Disable. 1: Enable. Any access to the floppy disk drive address ranges listed below reloads General Purpose Timer 1: — Primary floppy disk: I/O Port 3F2h-3F5h, 3F7h. — Secondary floppy disk: I/O Port 372h-375h, 377h. The active floppy disk drive is configured via F0 Index 93h[7].
0	**Re-trigger General Purpose Timer 1 on Primary Hard Disk Activity.** 0: Disable. 1: Enable. Any access to the primary hard disk address range selected in F0 Index 93h[5], reloads General Purpose Timer 1.

Index 8Ah	General Purpose Timer 2 Count Register (R/W)	Reset Value: 00h
7:0	**GPT2_COUNT.** This field represents the load value for General Purpose Timer 2. This value can represent either an 8-bit or 16-bit counter (configured in F0 Index 8Bh[5]). It is loaded into the counter when the timer is enabled (F0 Index 83h[1] = 1). Once the timer is enabled and a transition occurs on GPIO7, the timer is re-loaded. The counter is decremented with each clock of the configured timebase (1 msec or 1 sec selected at F0 Index 8Bh[3]). Upon expiration of the counter, an SMI is generated and the top level of status is F1BAR0+I/O Offset 00h/02h[9]. The second level of status is reported at F1BAR0+I/O Offset 04h/06h[1]). Once expired, this counter must be re-initialized by either disabling and enabling it, or by writing a new count value in this register. For GPIO7 to act as the reload for this counter, it must be enabled as such (F0 Index 8Bh[2]) and be configured as an input. (GPIO pin programming is at F0BAR0+I/O Offset 20h and 24h.)	

Index 8Bh	General Purpose Timer 2 Control Register (R/W)	Reset Value: 00h
7	**Re-trigger General Purpose Timer 1 (GP Timer 1) on Secondary Hard Disk Activity.** 0: Disable. 1: Enable. Any access to the secondary hard disk address range selected in F0 Index 93h[4] reloads GP Timer 1.	
6	**VGA Timer Base.** Selects timebase for VGA Timer Register (F0 Index 8Eh). 0: 1 millisecond. 1: 32 microseconds.	
5	**General Purpose Timer 2 (GP Timer 2) Shift.** GP Timer 2 is treated as an 8-bit or 16-bit timer. 0: 8-bit. The count value is loaded into GP Timer 2 Count Register (F0 Index 8Ah). 1: 16-bit. The value loaded into GP Timer 2 Count Register is shifted left by eight bits, the lower eight bits become zero, and this 16-bit value is used as the count for GP Timer 2.	
4	**General Purpose Timer 1 (GP Timer 1) Shift.** GP Timer 1 is treated as an 8-bit or 16-bit timer. 0: 8-bit. The count value is that loaded into GP Timer 1 Count Register (F0 Index 88h). 1: 16-bit. The value loaded into GP Timer 1 Count Register is shifted left by eight bit, the lower eight bits become zero, and this 16-bit value is used as the count for GP Timer 1.	
3	**General Purpose Timer 2 (GP Timer 2) Timebase.** Selects timebase for GP Timer 2 (F0 Index 8Ah). 0: 1 second. 1: 1 millisecond.	

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Re-trigger Timer on GPIO7 Pin Transition.** A rising-edge transition on the GPIO7 pin reloads GP Timer 2 (F0 Index 8Ah). 0: Disable. 1: Enable. For GPIO7 to work here, it must first be configured as an input. (GPIO pin programming is at F0BAR0+I/O Offset 20h and 24h.)
1:0	**Reserved.** Set to 0.
Index 8Ch	**IRQ Speedup Timer Count Register (R/W)** — **Reset Value: 00h**
7:0	**IRQ Speedup Timer Load Value.** This field represents the load value for the IRQ speedup timer. It is loaded into the counter when Suspend Modulation is enabled (F0 Index 96h[0] = 1) and an INTR or an access to I/O Port 061h occurs. When the event occurs, the Suspend Modulation logic is inhibited, permitting full performance operation of the GX1 module. Upon expiration, no SMI is generated; the Suspend Modulation begins again. The IRQ speedup timer's timebase is 1 msec. This speedup mechanism allows instantaneous response to system interrupts for full-speed interrupt processing. A typical value here would be 2 to 4 msec.
Index 8Dh	**Video Speedup Timer Count Register (R/W)** — **Reset Value: 00h**
7:0	**Video Speedup Timer Load Value.** This field represents the load value for the Video speedup timer. It is loaded into the counter when Suspend Modulation is enabled (F0 Index 96[0] = 1) and any access to the graphics controller occurs. When a video access occurs, the Suspend Modulation logic is inhibited, permitting full-performance operation of the GX1 module. Upon expiration, no SMI is generated, and Suspend Modulation begins again. The video speedup timer's timebase is 1 msec. This speedup mechanism allows instantaneous response to video activity for full speed during video processing calculations. A typical value here would be 50 msec to 100 msec.
Index 8Eh	**VGA Timer Count Register (R/W)** — **Reset Value: 00h**
7:0	**VGA Timer Load Value.** This field represents the load value for VGA Timer. It is loaded into the counter when the timer is enabled (F0 Index 83h[3] = 1). The counter is decremented with each clock of the configured timebase (F0 Index 8Bh[6]). Upon expiration of the counter, an SMI is generated and the status is reported at F1BAR0+I/O Offset 00h/02h[6] (only). Once expired, this counter must be re-initialized by either disabling and enabling it, or by writing a new count value in this register. **Note:** Although grouped with the power management Idle Timers, the VGA Timer is not a power management function. It is not affected by the Global Power Management Enable setting at F0 Index 80h[0].
Index 8Fh-92h	**Reserved** — **Reset Value: 00h**
Index 93h	**Miscellaneous Device Control Register (R/W)** — **Reset Value: 00h**
7	**Floppy Drive Port Select.** Indicates whether all system resources used to power manage the floppy drive use the primary, or secondary FDC addresses for decode. 0: Secondary. 1: Primary.
6	**Reserved.** Must be set to 1.
5	**Partial Primary Hard Disk Decode.** This bit is used to restrict the addresses which are decoded as primary hard disk accesses. 0: Power management monitors all reads and writes to I/O Port 1F0h-1F7h, 3F6h-3F7h (excludes writes to 3F7h), and 170h-177h, 376h-377h (excludes writes to 377h). 1: Power management monitors only writes to I/O Port 1F6h and 1F7h.
4	**Partial Secondary Hard Disk Decode.** This bit is used to restrict the addresses which are decoded as secondary hard disk accesses. 0: Power management monitors all reads and writes to I/O Port 170h-177h, 376h-377h (excludes writes to 377h). 1: Power management monitors only writes to I/O Port 176h and 177h.
3:2	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**Mouse on Serial Enable.** Mouse is present on a serial port. 0: No. 1: Yes. If a mouse is attached to a serial port (i.e., this bit is set to 1), that port is removed from the serial device list being used to monitor serial port access for power management purposes and added to the keyboard/mouse decode. This is done because a mouse, along with the keyboard, is considered an input device and is used only to determine when to blank the screen. This bit and bit 0 of this register determine the decode used for the Keyboard/Mouse Idle Timer Count Register (F0 Index 9Eh) as well as the Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch).
0	**Mouse Port Select.** Selects which serial port the mouse is attached to: 0: COM1. 1: COM2. For more information see the description of bit 1 in this register (above).
Index 94h-95h	**Suspend Modulation Register (R/W)** — **Reset Value: 0000h**
15:8	**Suspend Signal Asserted Counter.** This 8-bit counter represents the number of 32 µs intervals that the internal SUSP# signal is asserted to the GX1 module. Together with bits [7:0], perform the Suspend Modulation function for CPU power management. The ratio of SUSP# asserted-to-de-asserted sets up an effective (emulated) clock frequency, allowing the power manager to reduce GX1 module power consumption. This counter is prematurely reset if an enabled speedup event occurs (i.e., IRQ and video speedups).
7:0	**Suspend Signal De-asserted Counter.** This 8-bit counter represents the number of 32 µs intervals that the internal SUSP# signal is de-asserted to the GX1 module. Together with bits [15:8], perform the Suspend Modulation function for CPU power management. The ratio of SUSP# asserted-to-de-asserted sets up an effective (emulated) clock frequency, allowing the power manager to reduce GX1 module power consumption. This counter is prematurely reset if an enabled speedup event occurs (i.e., IRQ and video speedups).
Index 96h	**Suspend Configuration Register (R/W)** — **Reset Value: 00h**
7:3	**Reserved.** Must be set to 0.
2	**Suspend Mode Configuration.** Special 3V Suspend mode to support powering down the GX1 module during Suspend. 0: Disable. 1: Enable.
1	**SMI Speedup Configuration.** Selects how the Suspend Modulation function should react when an SMI occurs. 0: Use the IRQ Speedup Timer Count Register (F0 Index 8Ch) to temporarily disable Suspend Modulation when an SMI occurs. 1: Disable Suspend Modulation when an SMI occurs until a read to the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h). The purpose of this bit is to disable Suspend Modulation while the GX1 module is in the System Management Mode so that VSA and Power Management operations occur at full speed. Two methods for accomplishing this are: Map the SMI into the IRQ Speedup Timer Count Register (F0 Index 8Ch). - or - Have the SMI disable Suspend Modulation until the SMI handler reads the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h). This the preferred method. This bit has no affect if the Suspend Modulation feature is disabled (bit 0 = 0).
0	**Suspend Modulation Feature Enable.** This bit is used to enable/disable the Suspend Modulation feature. 0: Disable. 1: Enable. When enabled, the internal SUSP# signal is asserted and de-asserted for the durations programmed in the Suspend Modulation register (F0 Index 94h). The setting of this bit is mirrored in the Top Level PME/SMI Status register (F1BAR0+I/O Offset 00h/02h[15]. It is used by the SMI handler to determine if the SMI Speedup Disable register (F1BAR0+I/O Offset 08h) must be cleared on exit.
Index 97h	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 98h-99h	**Primary Hard Disk Idle Timer Count Register (Primary Channel) (R/W)** — **Reset Value: 0000h**
15:0	**Primary Hard Disk Idle Timer Count.** This idle timer is used to determine when the primary hard disk is not in use so that it can be powered down. The 16-bit value programmed here represents the period of hard disk inactivity after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the configured hard disk's data port (I/O port 1F0h or 3F6h). This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[0] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[0].
Index 9Ah-9Bh	**Floppy Disk Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Floppy Disk Idle Timer Count.** This idle timer is used to determine when the floppy disk drive is not in use so that it can be powered down. The 16-bit value programmed here represents the period of floppy disk drive inactivity after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the configured floppy drive's data port (I/O port 3F5h or 375h). This counter uses a 1 second time base. To enable this timer, set F0 Index 81h[1] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[1].
Index 9Ch-9Dh	**Parallel / Serial Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Parallel/Serial Idle Timer Count.** This idle timer is used to determine when the parallel and serial ports are not in use so that the ports can be power managed. The 16-bit value programmed in this register represents the period of inactivity for these ports after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the parallel (LPT) or serial (COM) I/O address spaces. If the mouse is enabled on a serial port, that port is not considered here. This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[2] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[2].
Index 9Eh-9Fh	**Keyboard / Mouse Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Keyboard / Mouse Idle Timer Count.** This idle timer determines when the keyboard and mouse are not in use so that the LCD screen can be blanked. The 16-bit value programmed in this register represents the period of inactivity for these ports after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to either the keyboard or mouse I/O address spaces (including the mouse serial port address space when a mouse is enabled on a serial port). This counter uses a 1 second time base. To enable this timer, set F0 Index 81h[3] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[3].
Index A0h-A1h	**User Defined Device 1 Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**User Defined Device 1 (UDEF1) Idle Timer Count.** This idle timer determines when the device configured as User Defined Device 1 (UDEF1) is not in use so that it can be power managed. The 16-bit value programmed in this register represents the period of inactivity for this device after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to memory or I/O address space configured in the F0 Index C0h (Base Address register) and F0 Index CCh (Control register). This counter uses a 1 second time base. To enable this timer, set F0 Index 81h[4] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[4].
Index A2h-A3h	**User Defined Device 2 Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**User Defined Device 2 (UDEF2) Idle Timer Count.** This idle timer determines when the device configured as UDEF2 is not in use so that it can be power managed. The 16-bit value programmed in this register represents the period of inactivity for this device after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to memory or I/O address space configured in the F0 Index C4h (Base Address register) and F0 Index CDh (Control register). This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[5] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[5].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index A4h-A5h	**User Defined Device 3 Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**User Defined Device 3 (UDEF3) Idle Timer Count.** This idle timer determines when the device configured as UDEF3 is not in use so that it can be power managed. The 16-bit value programmed in this register represents the period of inactivity for this device after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to memory or I/O address space configured in the UDEF3 Base Address Register (F0 Index C8h) and UDEF3 Control Register (F0 Index CEh). This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[6] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[6].
Index A6h-A7h	**Video Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Video Idle Timer Count.** This idle timer determines when the graphics subsystem has been idle as part of the Suspend-determination algorithm. The 16-bit value programmed in this register represents the period of video inactivity after which the system is alerted via an SMI. The count in this timer is automatically reset at any access to the graphics controller space. This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[7] = 1. Since the graphics controller is embedded in the GX1 module, video activity is communicated to the Core Logic module via the serial connection (PSERIAL register, bit 0). The Core Logic module also detects accesses to standard VGA space on PCI (3Bxh, 3Cxh, 3Dxh and A000h-B7FFh) if an external VGA controller is being used. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[7].
Index A8h-A9h	**Video Overflow Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Video Overflow Count.** Each time the video speedup counter is triggered, a 100 msec timer is started. If the 100 msec timer expires before the video speedup counter lapses, the Video Overflow Count register increments and the 100 msec timer retriggers. Software clears the overflow register when new evaluations are to begin. The count contained in this register can be combined with other data to determine the type of video accesses present in the system.
Index AAh-ABh	**Reserved** — **Reset Value: 00h**
Index ACh-ADh	**Secondary Hard Disk Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Secondary Hard Disk Idle Timer Count.** This idle timer is used to determine when the secondary hard disk is not in use so that it can be powered down. The 16-bit value programmed in this register represents the period of hard disk inactivity after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the configured hard disk's data port (I/O port 1F0h or 170h). This counter uses a 1 second timebase. To enable this timer, set F0 Index 83h[7] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[4].
Index AEh	**CPU Suspend Command Register (WO)** — **Reset Value: 00h**
7:0	**Software CPU Suspend Command.** If bit 0 in the Clock Stop Control register is set low (F0 Index BCh[0] = 0), a write to this register causes an internal SUSP#/SUSPA# handshake with the GX1 module, placing the GX1 module in a low-power state. The actual data written is irrelevant. Once in this state, any unmasked IRQ or SMI releases the GX1 module halt condition. If F0 Index BCh[0] = 1, writing to this register invokes a full system Suspend. In this case, the internal SUSP_3V signal is asserted after the SUSP#/SUSPA# halt. Upon a Resume event, the PLL delay programmed in the F0 Index BCh[7:4] is invoked, allowing the clock chip and GX1 module PLL to stabilize before de-asserting SUSP#.
Index AFh	**Suspend Notebook Command Register (WO)** — **Reset Value: 00h**
7:0	**Software CPU Stop Clock Suspend.** A write to this register causes a SUSP#/SUSPA# handshake with the CPU, placing the GX1 module in a low-power state. Following this handshake, the SUSP_3V signal is asserted. The SUSP_3V signal is intended to be used to stop all system clocks. Upon a Resume event, the internal SUSP_3V signal is de-asserted. After a slight delay, the Core Logic module de-asserts the SUSP# signal. Once the clocks are stable, the GX1 module de-asserts SUSPA# and system operation resumes.
Index B0h-B3h	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index B4h	**Floppy Port 3F2h Shadow Register (RO)** — **Reset Value: xxh**
7:0	**Floppy Port 3F2h Shadow.** Last written value of I/O Port 3F2h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B5h	**Floppy Port 3F7h Shadow Register (RO)** — **Reset Value: xxh**
7:0	**Floppy Port 3F7h Shadow.** Last written value of I/O Port 3F7h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B6h	**Floppy Port 372h Shadow Register (RO)** — **Reset Value: xxh**
7:0	**Floppy Port 372h Shadow.** Last written value of I/O Port 372h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B7h	**Floppy Port 377h Shadow Register (RO)** — **Reset Value: xxh**
7:0	**Floppy Port 377h Shadow.** Last written value of I/O Port 377h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B8h	**DMA Shadow Register (RO)** — **Reset Value: xxh**
7:0	**DMA Shadow.** This 8-bit port sequences through the following list of shadowed DMA Controller registers. At power on, a pointer starts at the first register in the list and continuing through the other registers in subsequent reads according to the read sequence. A write to this register resets the read sequence to the first register. Each shadow register in the sequence contains the last data written to that location. The read sequence for this register is: 1. DMA Channel 0 Mode Register 2. DMA Channel 1 Mode Register 3. DMA Channel 2 Mode Register 4. DMA Channel 3 Mode Register 5. DMA Channel 4 Mode Register 6. DMA Channel 5 Mode Register 7. DMA Channel 6 Mode Register 8. DMA Channel 7 Mode Register 9. DMA Channel Mask Register (bit 0 is channel 0 mask, etc.) 10. DMA Busy Register (bit 0 or 1 means a DMA occurred within last 1 msec, all other bits are 0)
Index B9h	**PIC Shadow Register (RO)** — **Reset Value: xxh**
7:0	**PIC Shadow.** This 8-bit port sequences through the following list of shadowed Interrupt Controller registers. At power on, a pointer starts at the first register in the list and continuing through the other registers in subsequent reads according to the read sequence. A write to this register resets the read sequence to the first register. Each shadow register in the sequence contains the last data written to that location. The read sequence for this register is: 1. PIC1 ICW1 2. PIC1 ICW2 3. PIC1 ICW3 4. PIC1 ICW4 - Bits [7:5] of ICW4 are always 0. 5. PIC1 OCW2 - Bits [6:3] of OCW2 are always 0 (See Note). 6. PIC1 OCW3 - Bits [7:4] are 0 and bits [6:3] are 1. 7. PIC2 ICW1 8. PIC2 ICW2 9. PIC2 ICW3 10. PIC2 ICW4 - Bits [7:5] of ICW4 are always 0. 11. PIC2 OCW2 - Bits [6:3] of OCW2 are always 0 (See Note). 12. PIC2 OCW3 - Bits [7:4] are 0 and bits [6:3] are 1. **Note:** To restore OCW2 to the shadow register value, write the appropriate address twice. First with the shadow register value, then with the shadow register value ORed with C0h.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index BAh	**PIT Shadow Register (RO)** — **Reset Value: xxh**
7:0	**PIT Shadow.** This 8-bit port sequences through the following list of shadowed Programmable Interval Timer registers. At power on, a pointer starts at the first register in the list and continuing through the other registers in subsequent reads according to the read sequence. A write to this register resets the read sequence to the first register. Each shadow register in the sequence contains the last data written to that location. The read sequence for this register is: 1. Counter 0 LSB (least significant byte) 2. Counter 0 MSB 3. Counter 1 LSB 4. Counter 1 MSB 5. Counter 2 LSB 6. Counter 2 MSB 7. Counter 0 Command Word 8. Counter 1 Command Word 9. Counter 2 Command Word **Note:** The LSB/MSB of the count is the Counter base value, not the current value. Bits [7:6] of the command words are not used.
Index BBh	**RTC Index Shadow Register (RO)** — **Reset Value: xxh**
7:0	**RTC Index Shadow.** The RTC Shadow register contains the last written value of the RTC Index register (I/O Port 070h).
Index BCh	**Clock Stop Control Register (R/W)** — **Reset Value: 00h**
7:4	**PLL Delay.** The programmed value in this field sets the delay (in milliseconds) after a break event occurs before the internal SUSP# signal is de-asserted to the GX1 module. This delay is designed to allow the clock chip and CPU PLL to stabilize before starting execution. This delay is only invoked if the STP_CLK bit was set. The 4-bit field allows values from 0 to 15 msec. 0000: 0 msec — 0100: 4 msec — 1000: 8 msec — 1100: 12 msec 0001: 1 msec — 0101: 5 msec — 1001: 9 msec — 1101: 13 msec 0010: 2 msec — 0110: 6 msec — 1010: 10 msec — 1110: 14 msec 0011: 3 msec — 0111: 7 msec — 1011: 11 msec — 1111: 15 msec
3:1	**Reserved.** Set to 0.
0	**CPU Clock Stop.** 0: Normal internal SUSP#/SUSPA# handshake. 1: Full system Suspend.
Note:	This register configures the Core Logic module to support a 3V Suspend mode. Setting bit 0 causes the SUSP_3V signal to assert after the appropriate conditions, stopping the system clocks. A delay of 0-15 msec is programmable (bits [7:4]) to allow for a delay for the clock chip and CPU PLL to stabilize when an event Resumes the system. A write to the CPU Suspend Command register (F0 Index AEh) with bit 0 written as: 0: Internal SUSP#/SUSPA# handshake occurs. The GX1 module is put into a low-power state, and the system clocks are not stopped. When a break/resume event occurs, it releases the CPU halt condition. 1: Internal SUSP#/SUSPA# handshake occurs and the SUSP_3V signal is asserted, thus invoking a full system Suspend (both GX1 module and system clocks are stopped). When a break event occurs, the SUSP_3V signal is de-asserted, the PLL delay programmed in bits [7:4] are invoked which allows the clock chip and GX1 module PLL to stabilize before de-asserting the internal SUSP# signal.
Index BDh-BFh	**Reserved** — **Reset Value: 00h**
Index C0h-C3h	**User Defined Device 1 Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**User Defined Device 1 Base Address.** This 32-bit register supports power management (Trap and Idle timer resources) for a PCMCIA slot or some other device in the system. The value in this register is used as the address comparator for the device trap/timer logic. The device can be memory or I/O mapped (configured in F0 Index CCh). The Core Logic module cannot snoop addresses on the Fast-PCI bus unless it actually claims the cycle. Therefore, Traps and Idle timers cannot support power management of devices on the Fast-PCI bus.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index C4h-C7h	**User Defined Device 2 Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**User Defined Device 2 Base Address.** This 32-bit register supports power management (Trap and Idle timer resources) for a PCMCIA slot or some other device in the system. The value in this register is used as the address comparator for the device trap/timer logic. The device can be memory or I/O mapped (configured in F0 Index CDh). The Core Logic module cannot snoop addresses on the Fast-PCI bus unless it actually claims the cycle. Therefore, Traps and Idle timers cannot support power management of devices on the Fast-PCI bus.
Index C8h-CBh	**User Defined Device 3 Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**User Defined Device 3 Base Address.** This 32-bit register supports power management (Trap and Idle timer resources) for a PCMCIA slot or some other device in the system. The value in this register is used as the address comparator for the device trap/timer logic. The device can be memory or I/O mapped (configured in F0 Index CEh). The Core Logic module cannot snoop addresses on the Fast-PCI bus unless the it actually claims the cycle. Therefore, Traps and Idle timers cannot support power management of devices on the Fast-PCI bus.
Index CCh	**User Defined Device 1 Control Register (R/W)** — **Reset Value: 00h**
7	**Memory or I/O Mapped.** Determines how User Defined Device 1 is mapped. 0: I/O. 1: Memory.
6:0	**Mask.** If bit 7 = 0 (I/O): Bit 6 0: Disable write cycle tracking 1: Enable write cycle tracking Bit 5 0: Disable read cycle tracking 1: Enable read cycle tracking Bits [4:0] Mask for address bits A[4:0] If bit 7 = 1 (Memory): Bits [6:0] Mask for address memory bits A[15:9] (512 bytes min. and 64 KB max.) A[8:0] are ignored. **Note:** A "1" in a mask bit means that the address bit is ignored for comparison.
Index CDh	**User Defined Device 2 Control Register (R/W)** — **Reset Value: 00h**
7	**Memory or I/O Mapped.** determines how User Defined Device 2 is mapped. 0: I/O 1: Memory
6:0	**Mask.** If bit 7 = 0 (I/O): Bit 6 0: Disable write cycle tracking 1: Enable write cycle tracking Bit 5 0: Disable read cycle tracking 1: Enable read cycle tracking Bits [4:0] Mask for address bits A[4:0] If bit 7 = 1 (Memory): Bits [6:0] Mask for address memory bits A[15:9] (512 bytes min. and 64 KB max.) A[8:0] are ignored. **Note:** A "1" in a mask bit means that the address bit is ignored for comparison.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index CEh	**User Defined Device 3 Control Register (R/W)** — **Reset Value: 00h**
7	**Memory or I/O Mapped.** Determines how User Defined Device 3 is mapped. 0: I/O. 1: Memory.
6:0	**Mask.** If bit 7 = 0 (I/O): Bit 6 0: Disable write cycle tracking 1: Enable write cycle tracking Bit 5 0: Disable read cycle tracking 1: Enable read cycle tracking Bits [4:0] Mask for address bits A[4:0] If bit 7 = 1 (Memory): Bits [6:0] Mask for address memory bits A[15:9] (512 bytes min. and 64 KB max.) A[8:0] are ignored. **Note:** A "1" in a mask bit means that the address bit is ignored for comparison.
Index CFh	**Reserved** — **Reset Value: 00h**
Index D0h	**Software SMI Register (WO)** — **Reset Value: 00h**
7:0	**Software SMI.** A write to this location generates an SMI. The data written is irrelevant. This register allows software entry into SMM via normal bus access instructions.
Index D1h-EBh	**Reserved** — **Reset Value: 00h**
Index ECh	**Timer Test Register (R/W)** — **Reset Value: 00h**
7:0	**Timer Test Value.** The Timer Test register is intended only for test and debug purposes. It is not intended for setting operational timebases. For normal operation, never write to this register.
Index EDh-F3h	**Reserved** — **Reset Value: 00h**
Index F4h	**Second Level PME/SMI Status Register 1 (RC)** — **Reset Value: 00h** The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F0 Index 84h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F0 Index 84h can be read instead.
7:3	**Reserved.** Reads as 0.
2	**GPWIO2 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO2 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO2 is enabled as an input: F1BAR1+I/O Offset 15h[2] = 0. 2) Set F1BAR1+I/O Offset 15h[6] = 1 to allow SMI generation.
1	**GPWIO1 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO1 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO1 is enabled as an input: F1BAR1+I/O Offset 15h[1] = 0. 2) Set F1BAR1+I/O Offset 15h[5] to 1 to allow SMI generation.
0	**GPWIO0 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO0 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO0 is enabled as an input: F1BAR1+I/O Offset 15h[0] = 0. 2) Set F1BAR1+I/O Offset 15h[4] to 1 to allow SMI generation.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index F5h	**Second Level PME/SMI Status Register 2 (RC)** **Reset Value: 00h**
	The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F0 Index 85h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F0 Index 85h can be read instead.
7	**Video Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Video Idle Timer Count Register, (F0 Index A6h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[7] = 1.
6	**User Defined Device Idle Timer 3 (UDEF3) SMI Status.** Indicates whether or not an SMI was caused by expiration of User Defined Device 3 (UDEF3) Idle Timer Count Register (F0 Index A4h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[6] = 1.
5	**User Defined Device Idle Timer 2 (UDEF2) SMI Status.** Indicates whether or not an SMI was caused by expiration of User Defined Device 2 (UDEF2) Idle Timer Count Register (F0 Index A2h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[5] = 1.
4	**User Defined Device Idle Timer 1 (UDEF1) SMI Status.** Indicates whether or not an SMI was caused by expiration of User Defined Device 1 (UDEF1) Idle Timer Count Register (F0 Index A0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[4] = 1.
3	**Keyboard/Mouse Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Keyboard/ Mouse Idle Timer Count Register (F0 Index 9Eh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[3] = 1.
2	**Parallel/Serial Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[2] = 1.
1	**Floppy Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Floppy Disk Idle Timer Count Register (F0 Index 9Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[1] = 1.
0	**Hard Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Hard Disk Idle Timer Count Register (F0 Index 98h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[0] = 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index F6h	**Second Level PME/SMI Status Register 3 (RC)** **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F0 Index 86h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F0 Index 86h can be read instead.	
7	**Video Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the Video I/O Trap. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[7] = 1.
6	**Reserved.** Reads as 0.
5	**Secondary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the secondary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[6] = 1.
4	**Secondary Hard Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Secondary Hard Disk Idle Timer Count register (F0 Index ACh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[7] = 1.
3	**Keyboard/Mouse Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the keyboard or mouse. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[3] = 1.
2	**Parallel/Serial Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to either the serial or parallel ports. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[2] =1.
1	**Floppy Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the floppy disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[1] = 1.
0	**Primary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the primary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[0] = 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description	
Index F7h	**Second Level PME/SMI Status Register 4 (RC)**	**Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels except for bit 7 which has a third level of status reporting at F0BAR0+I/O 0Ch/1Ch. A read-only “Mirror” version of this register exists at F0 Index 87h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F0 Index 87h can be read instead.		
7	**GPIO Event SMI Status (Read Only, Read does not Clear).** Indicates whether or not an SMI was caused by a transition of any of the GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] = 0. F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled to generate a PME and setting F1BAR1+I/O Offset 0Ch[0] = 0 enables the PME to generate an SMI. In addition, the selected GPIO must be enabled as an input (F0BAR0+I/O Offset 20h and 24h). The next level (third level) of SMI status is at F0BAR0+I/O 0Ch/1Ch.	
6	**Thermal Override SMI Status.** Indicates whether or not an SMI was caused by an assertion of the THRM#. 0: No. 1: Yes. To enable SMI generation set F0 Index 83h[4] = 1.	
5:4	**Reserved.** Read as 0.	
3	**SIO PWUREQ SMI Status.** Indicates whether or not an SMI was caused by a power-up event from the SIO. 0: No. 1: Yes. A power-up event is defined as any of the following events/activities: — RI2#. — SDATA_IN2. — IRRX1 (CEIR). To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] = 0.	
2	**Codec SDATA_IN SMI Status.** Indicates whether or not an SMI was caused by AC97 Codec producing a positive edge on SDATA_IN. 0: No. 1: Yes. To enable SMI generation, set F0 Index 80h[5] = 1.	
1	**RTC Alarm (IRQ8#) SMI Status.** Indicates whether or not an SMI was caused by an RTC interrupt. 0: No. 1: Yes. This SMI event can only occur while in 3V Suspend and an RTC interrupt occurs and F1BAR1+I/O Offset 0Ch[0] = 0.	
0	**ACPI Timer SMI Status.** Indicates whether or not an SMI was caused by an ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch) MSB toggle. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[5] = 1.	
Index F8h-FFh	**Reserved**	**Reset Value: 00h**

Core Logic Module (Continued)

5.4.1.1 GPIO Support Registers

F0 Index 10h, Base Address Register 0 (F0BAR0) points to the base address of where the GPIO runtime and configuration registers are located. Table 5-29 gives the bit formats of I/O mapped registers accessed through F0BAR0.

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers

Bit	Description
Offset 00h-03h	**GPDO0 — GPIO Data Out 0 Register (R/W)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data Out.** Bits [31:0] of this register correspond to GPIO31-GPIO0 signals, respectively. The value of each bit determines the value driven on the corresponding GPIO signal when its output buffer is enabled. Writing to the bit latches the written data unless the bit is locked by the GPIO Configuration register Lock bit (F0BAR0+I/O Offset 24h[3]). Reading the bit returns the value, regardless of the signal value and configuration. 0: Corresponding GPIO signal is driven to low when output enabled. 1: Corresponding GPIO signal is driven or released to high (according to buffer type and static pull-up selection) when output is enabled.
Offset 04h-07h	**GPDI0 — GPIO Data In 0 Register (RO)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data In.** Bits [31:0] of this register correspond to GPIO31-GPIO0 signals, respectively. Reading each bit returns the value of the corresponding GPIO signal, regardless of the signal configuration and the GPDO0 register (F0BAR0+I/O Offset 00h) value. Writes to this register are ignored. 0: Corresponding GPIO signal level is low. 1: Corresponding GPIO signal level is high.
Offset 08h-0Bh	**GPIEN0 — GPIO Interrupt Enable 0 Register (R/W)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Power Management Event (PME) Enable.** Bits [15:0] correspond to GPIO15-GPIO0 signals, respectively. Each bit allows PME generation by the corresponding GPIO signal. 0: Disable PME generation. 1: Enable PME generation. **Notes:** 1) All of the enabled GPIO PMEs are always reported at F1BAR1+I/O Offset 10h[3]. 2) Any enabled GPIO PME can be selected to generate an SCI or SMI at F1BAR1+I/O Offset 0Ch[0]. If SCI is selected, then the individually selected GPIO PMEs are globally enabled for SCI generation at F1BAR1+I/O Offset 12h[3] and the status is reported at F1BAR1+I/O Offset 10h[3]. If SMI is selected, the individually selected GPIO PMEs generate an SMI and the status is reported at F1BAR0+I/O Offset 00h/02h[0].
Offset 0Ch-0Fh	**GPST0 — GPIO Status 0 Register (R/W1C)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Status**. Bits [15:0] correspond to GPIO15-GPIO0 signals, respectively. Each bit reports a 1 when hardware detects the edge (rising/falling on the GPIO signal) that is programmed in F0BAR0+I/O Offset 24h[5]. If the corresponding bit in F0BAR0+I/O Offset 08h is set, this edge generates a PME. 0: No active edge detected since the bit was last cleared. 1: Active edge detected. Writing 1 to the a Status bit clears it to 0. This is the third level of SMI status reporting to the second level at F0 Index 87h/F7h[7] and the top level at F1BAR0+I/O Offset 00h/02h[0]. Clearing the third level also clears the second and top levels. This is the second level of SCI status reporting to the top level at F1BAR1+Offset 10h[3]. The status must be cleared at both the this level and the top level (i.e., the top level is not automatically cleared when a bit in this register is cleared).
Offset 10h-13h	**GPDO1 — GPIO Data Out 1 Register (R/W)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data Out.** Bits [31:0] of this register correspond to GPIO63-GPIO32 signals, respectively. The value of each bit determines the value driven on the corresponding GPIO signal when its output buffer is enabled. Writing to the bit latches the written data unless the bit is locked by the GPIO Configuration register Lock bit (F0BAR0+I/O Offset 24h[3]). Reading the bit returns the value, regardless of the signal value and configuration. 0: Corresponding GPIO signal driven to low when output enabled. 1: Corresponding GPIO signal driven or released to high (according to buffer type and static pull-up selection) when output enabled.

Core Logic Module (Continued)

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers (Continued)

Bit	Description
Offset 14h-17h	**GPDI1 — GPIO Data In 1 Register (RO)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data In.** Bits [31:0] of this register correspond to GPIO63-GPIO32 signals, respectively. Reading each bit returns the value of the corresponding GPIO signal, regardless of the signal configuration and the GPDO1 register (F0BAR0+I/O Offset 10h) value. Writes to this register are ignored. 0: Corresponding GPIO signal level low. 1: Corresponding GPIO signal level high.
Offset 18h-1Bh	**GPIEN1 — GPIO Interrupt Enable 1 Register (R/W)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Power Management Event (PME) Enable.** Bits [15:0] of this register correspond to GPIO47-GPIO32 signals, respectively. Each bit allows PME generation by the corresponding GPIO signal. 0: Disable PME generation. 1: Enable PME generation. **Notes:** 1) All of the enabled GPIO PMEs are always reported at F1BAR1+I/O Offset 10h[3]. 2) Any enabled GPIO PME can be selected to generate an SCI or SMI at F1BAR1+I/O Offset 0Ch[0]. If SCI is selected, the individually selected GPIO PMEs are globally enabled for SCI generation at F1BAR1+I/O Offset 12h[3] and the status is reported at F1BAR1+I/O Offset 10h[3]. If SMI is selected, the individually selected GPIO PMEs generate an SMI and the status is reported at F1BAR0+I/O Offset 00h/02h[0].
Offset 1Ch-1Fh	**GPST1 — GPIO Status 1 Register (R/W1C)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Status**. Bits [15:0] correspond to GPIO47-GPIO32 signals, respectively. Each bit reports a 1 when hardware detects the edge (rising/falling on the GPIO signal) that is programmed in F0BAR0+I/O Offset 24h[5]. If the corresponding bit in F0BAR0+I/O Offset 18h is set, this edge generates a PME. 0: No active edge detected since the bit was last cleared. 1: Active edge detected. Writing 1 to the a Status bit clears it to 0. This is the third level of SMI status reporting to the second level at F0 Index 87h/F7h[7] and the top level at F1BAR0+I/O Offset 00h/02h[0]. Clearing the third level also clears the second and top levels. This is the second level of SCI status reporting to the top level at F1BAR1+Offset 10h[3]. The status must be cleared at both the this level and the top level (i.e., the top level is not automatically cleared when a bit in this register is cleared).
Offset 20h-23h	**GPIO Signal Configuration Select Register (R/W)** **Reset Value: 00000000h**
31:6	**Reserved.** Must be set to 0.

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers (Continued)

Bit	Description
5:0	**Signal Select.** Selects the GPIO signal to be configured in the Bank selected via bit 5 setting (i.e., Bank 0 or Bank 1). See Table 3-2 on page 81 for GPIO ball muxing options. GPIOs without an associated ball number are not available externally. **Bank 0** 000000 = GPIO0 (EBGA: H1 / TEPBGA: D11) 000001 = GPIO1 (EBGA: H2, AL12 / TEPBGA: D10, N30) 000010 = GPIO2 000011 = GPIO3 000100 = GPIO4 000101 = GPIO5 000110 = GPIO6 (EBGA: AH3 / TEPBGA: D28) 000111 = GPIO7 (EBGA: AH4 / TEPBGA: C30) 001000 = GPIO8 (EBGA: AJ2 / TEPBGA: C31) 001001 = GPIO9 (EBGA: AG4 / TEPBGA: C28) 001010 = GPIO10 (EBGA: AJ1 / TEPBGA: B29) 001011 = GPIO11 (EBGA: H30 / TEPBGA: AJ8) 001100 = GPIO12 (EBGA: AJ12 / TEPBGA: N29) 001101 = GPIO13 (EBGA: AL11 / TEPBGA: M29) 001110 = GPIO14 (EBGA: F1 / TEPBGA: D9) 001111 = GPIO15 (EBGA: G3 / TEPBGA: A8) 010000 = GPIO16 (EBGA: AL15 / TEPBGA: V31) 010001 = GPIO17 (EBGA: J4 / TEPBGA: A10) 010010 = GPIO18 (EBGA: A28 / TEPBGA: AG1) 010011 = GPIO19 (EBGA: H4 / TEPBGA: C9) 010100 = GPIO20 (EBGA: H3, AJ13 / TEPBGA: A9, N31) 010101 = GPIO21 010110 = GPIO22 010111 = GPIO23 011000 = GPIO24 011001 = GPIO25 011010 = GPIO26 011011 = GPIO27 011100 = GPIO28 011101 = GPIO29 011110 = GPIO30 011111 = GPIO31 **Bank 1** 100000 = GPIO32 (EBGA: AJ11 / TEPBGA: M28) 100001 = GPIO33 (EBGA: AL10 / TEPBGA: L31) 100010 = GPIO34 (EBGA: AK10 / TEPBGA: L30) 100011 = GPIO35 (EBGA: AJ10 / TEPBGA: L29) 100100 = GPIO36 (EBGA: AL9 / TEPBGA: L28) 100101 = GPIO37 (EBGA: AK9 / TEPBGA: K31) 100110 = GPIO38 (EBGA: AJ9 / TEPBGA: K28) 100111 = GPIO39 (EBGA: AL8 / TEPBGA: J31) 101000 = GPIO40 (EBGA: A21 / TEPBGA: Y3) 101001 = GPIO41 (EBGA: C19 / TEPBGA: W4) 101010 = GPIO42 101011 = GPIO43 101100 = GPIO44 101101 = GPIO45 101110 = GPIO46 101111 = GPIO47 110000 = GPIO48 110001 = GPIO49 110010 = GPIO50 110011 = GPIO51 110100 = GPIO52 110101 = GPIO53 110110 = GPIO54 110111 = GPIO55 111000 = GPIO56 111001 = GPIO57 111010 = GPIO58 111011 = GPIO59 111100 = GPIO60 111101 = GPIO61 111110 = GPIO62 111111 = GPIO63 (Note) **Note:** GPIO63 can be used to generate the PWRBTN# input signal. See PWRBTN# signal description in Section 2.4.15 "Power Management Interface Signals" on page 74.

Offset 24h-27h **GPIO Signal Configuration Access Register (R/W)** **Reset Value: 00000044h**

This register is used to indicate configuration for the GPIO signal that is selected in the GPIO Signal Configuration Select Register (above).

Note: PME debouncing, polarity, and edge/level configuration is only applicable on GPIO0-GPIO15 signals (Bank 0 = 00000 to 01111) and on GPIO32-GPIO47 signals (Bank 1 settings of 00000 to 01111). The remaining GPIOs (GPIO16-GPIO31 and GPIO48-GPIO63) can not generate PMEs, therefore these bits have no function and read 0.

31:7	**Reserved.** Must be set to 0.
6	**PME Debounce Enable.** Enables/disables IRQ debounce (debounce period = 16 ms). 0: Disable. 1: Enable. (Default) See the note in the description of this register for more information about the default value of this bit.
5	**PME Polarity.** Selects the polarity of the signal that issues a PME from the selected GPIO signal (falling/low or rising/high). 0: Falling edge or low level input. (Default) 1: Rising edge or high level input. See the note in the description of this register for more information about the default value of this bit.

Core Logic Module (Continued)

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers (Continued)

Bit	Description
4	**PME Edge/Level Select.** Selects the type (edge or level) of the signal that issues a PME from the selected GPIO signal. 0: Edge input. (Default) 1: Level input. For normal operation, always set this bit to 0 (edge input). Erratic system behavior results if this bit is set to 1. See the note in the description of this register for more information about the default value of this bit.
3	**Lock.** This bit locks the selected GPIO signal. Once this bit is set to 1 by software, it can only be cleared to 0 by power on reset or by WATCHDOG reset. 0: No effect. (Default) 1: Direction, output type, pull-up and output value locked.
2	**Pull-Up Control.** Enables/disables the internal pull-up capability of the selected GPIO signal. It supports open-drain output signals with internal pull-ups and TTL input signals. 0: Disable. 1: Enable. (Default) Bits [1:0] of this register must = 01 for this bit to have effect.
1	**Output Type.** Controls the output buffer type (open-drain or push-pull) of the selected GPIO signal. 0: Open-drain. (Default) 1: Push-pull. Bit 0 of this register must be set to 1 for this bit to have effect.
0	**Output Enable.** Indicates the GPIO signal output state. It has no effect on input. 0: TRI-STATE - Setting for GPIO to function as an input only. (Default) 1: Output enabled.
Offset 28h-2Bh	**GPIO Reset Control Register (R/W)** — **Reset Value: 00000000h**
31:1	**Reserved.** Must be set to 0.
0	**GPIO Reset.** Reset the GPIO logic. 0: Disable. 1: Enable. Write 0 to clear. This bit is level-sensitive and must be cleared after the reset is enabled (normal operation requires this bit to be 0).

Core Logic Module (Continued)

5.4.1.2 LPC Support Registers

F0 Index 14h, Base Address Register 1 (F0BAR1) points to the base address of the register space that contains the configuration registers for LPC support. Table 5-31 gives the bit formats of the I/O mapped registers accessed through F0BAR1.

The LPC Interface supports all features described in the LPC bus specification 1.0, with the following exceptions:

- Only 8- or 16-bit DMA, depending on channel number. Does not support the optional larger transfer sizes.
- Only one external DRQ pin.

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers

Bit	Description
Offset 00h-03h	**SERIRQ_SRC — Serial IRQ Source Register (R/W)** **Reset Value: 00000000h**
31:21	**Reserved.**
20	**INTD Source.** Selects the interface source of the INTD# signal. 0: PCI - INTD#. (Program PMR[24] = 1 to enable INTD# function on ball.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
19	**INTC Source.** Selects the interface source of the INTC# signal. 0: PCI - INTC#. (Program PMR[4] =1 and PMR[9] = 0 to enable INTD# function on ball. 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
18	**INTB Source.** Selects the interface source of the INTB# signal. 0: PCI - INTB#. 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
17	**INTA Source.** Selects the interface source of the INTA# signal. 0: PCI - INTA#. 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
16	**Reserved.** Set to 0.
15	**IRQ15 Source.** Selects the interface source of the IRQ15 signal. 0: ISA - IRQ15. (Program PMR[18] = 0 and PMR[8] = 1 to enable IRQ15 function on ball.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
14	**IRQ14 Source.** Selects the interface source of the IRQ14 signal. 0: ISA - IRQ14. (Program PMR[24] = 0 to enable IRQ14 function on ball.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
13	**IRQ13 Source.** Selects the interface source of the internal IRQ13 signal. 0: ISA - IRQ13. (Internal signal - An input from the CPU indicating that a floating point error has been detected and that internal INTR should be asserted.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
12	**IRQ12 Source.** Selects the interface source of the IRQ12 signal. 0: ISA - IRQ12. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
11	**IRQ11 Source.** Selects the interface source of the IRQ11 signal. 0: ISA - IRQ11. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
10	**IRQ10 Source.** Selects the interface source of the IRQ10 signal. 0: ISA - IRQ10. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
9	**IRQ9 Source.** Selects the interface source of the IRQ9 signal. 0: ISA - IRQ9. (Program PMR[24] = 1.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
8	**IRQ8# Source.** Selects the interface source of the IRQ8# signal. 0: ISA - IRQ8#. (Internal signal - Connected to internal RTC.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
7	**IRQ7 Source.** Selects the interface source of the IRQ7 signal. 0: ISA - IRQ7. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
6	**IRQ6 Source.** Selects the interface source of the IRQ6 signal. 0: ISA - IRQ6. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
5	**IRQ5 Source.** Selects the interface source of the IRQ5 signal. 0: ISA - IRQ5. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
4	**IRQ4 Source.** Selects the interface source of the IRQ4 signal. 0: ISA - IRQ4. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
3	**IRQ3 Source.** Selects the interface source of the IRQ3 signal. 0: ISA - IRQ3. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
2	**Reserved.** Must be set to 0.
1	**IRQ1 Source.** Selects the interface source of the IRQ1 signal. 0: ISA - IRQ1. (Unavailable externally.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
0	**IRQ0 Source.** Selects the interface source of the IRQ0 signal. 0: ISA - IRQ0. (Internal signal - Connected to OUT0, System Timer, of the internal 8254 PIT.) 1: LPC - SERIRQ. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable SERIRQ function on ball.)
Offset 04h-07h	**SERIRQ_LVL — Serial IRQ Level Control Register (R/W)** **Reset Value: 00000000h**
31:21	**Reserved**
20	**INTD# Polarity.** If LPC is selected as the interface source for INTD# (F0BAR1+I/O Offset 00h[20] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
19	**INTC# Polarity.** If LPC is selected as the interface source for INTC# (F0BAR1+I/O Offset 00h[19] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
18	**INTB# Polarity.** If LPC is selected as the interface source for INTB# (F0BAR1+I/O Offset 00h[18] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
17	**INTA# Polarity.** If LPC is selected as the interface source for INTA# (F0BAR1+I/O Offset 00h[17] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
16	**Reserved.** Must be set to 0.
15	**IRQ15 Polarity.** If LPC is selected as the interface source for IRQ15 (F0BAR1+I/O Offset 00h[15] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
14	**IRQ14 Polarity.** If LPC is selected as the interface source for IRQ14 (F0BAR1+I/O Offset 00h[14] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
13	**IRQ13 Polarity.** If LPC is selected as the interface source for IRQ13 (F0BAR1+I/O Offset 00h[13] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
12	**IRQ12 Polarity.** If LPC is selected as the interface source for IRQ12 (F0BAR1+I/O Offset 00h[12] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
11	**IRQ11 Polarity.** If LPC is selected as the interface source for IRQ11 (F0BAR1+I/O Offset 00h[11] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
10	**IRQ10 Polarity.** If LPC is selected as the interface source for IRQ10 (F0BAR1+I/O Offset 00h[10] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
9	**IRQ9 Polarity.** If LPC is selected as the interface source for IRQ9 (F0BAR1+I/O Offset 00h[9] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
8	**IRQ8# Polarity.** If LPC is selected as the interface source for IRQ8# (F0BAR1+I/O Offset 00h[8] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
7	**IRQ7 Polarity.** If LPC is selected as the interface source for IRQ7 (F0BAR1+I/O Offset 00h[7] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
6	**IRQ6 Polarity.** If LPC is selected as the interface source for IRQ6 (F0BAR1+I/O Offset 00h[6] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
5	**IRQ5 Polarity.** If LPC is selected as the interface source for IRQ5 (F0BAR1+I/O Offset 00h[5] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
4	**IRQ4 Polarity.** If LPC is selected as the interface source for IRQ4 (F0BAR1+I/O Offset 00h[4] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
3	**IRQ3 Polarity.** If LPC is selected as the interface source for IRQ3 (F0BAR1+I/O Offset 00h[3] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
2	**SMI# Polarity.** This bit allows signal polarity selection of the SMI# generated from LPC. 0: Active high. 1: Active low.
1	**IRQ1 Polarity.** If LPC is selected as the interface source for IRQ1 (F0BAR1+I/O Offset 00h[1] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
0	**IRQ0 Polarity.** If LPC is selected as the interface source for IRQ0 (F0BAR1+I/O Offset 00h[0] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
Offset 08h-0Bh	**SERIRQ_CNT — Serial IRQ Control Register (R/W)** **Reset Value: 00000000h**
31:8	**Reserved.**
7	**Serial IRQ Enable.** 0: Disable. 1: Enable.
6	**Serial IRQ Interface Mode.** 0: Continuous. 1: Quiet.
5:2	**Number of IRQ Data Frames.** 0000: 17 frames 0100: 21 frames 1000: 25 frames 1100: 29 frames 0001: 18 frames 0101: 22 frames 1001: 26 frames 1101: 30 frames 0010: 19 frames 0110: 23 frames 1010: 27 frames 1110: 31 frames 0011: 20 frames 0111: 24 frames 1011: 28 frames 1111: 32 frames
1:0	**Start Frame Pulse Width.** 00: 4 Clocks 01: 6 Clocks 10: 8 Clocks 11: Reserved
Offset 0Ch-0Fh	**DRQ_SRC — DRQ Source Register (R/W)** **Reset Value: 00000000h** DRQx are internal signals between the Core Logic and SuperI/O modules.
31:8	**Reserved.**
7	**DRQ7 Source.** Selects the interface source of the DRQ7 signal. 0: ISA - DRQ7. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)
6	**DRQ6 Source.** Selects the interface source of the DRQ6 signal. 0: ISA - DRQ6. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)
5	**DRQ5 Source.** Selects the interface source of the DRQ5 signal. 0: ISA - DRQ5. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)
4	**LPC BM0 Cycles.** Allow LPC Bus Master 0 Cycles. 0: Enable. 1: Disable.
3	**DRQ3 Source.** Selects the interface source of the DRQ3 signal. 0: ISA - DRQ3. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)
2	**DRQ2 Source.** Selects the interface source of the DRQ2 signal. 0: ISA - DRQ2. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)
1	**DRQ1 Source.** Selects the interface source of the DRQ1 signal. 0: ISA - DRQ1. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)
0	**DRQ0 Source.** Selects the interface source of the DRQ0 signal. 0: ISA - DRQ0. (Unavailable externally.) 1: LPC - LDRQ#. (Program PMR[14] = 1 and PMR[22] = 1 or LPC_ROM strap = 1 to enable LDRQ# function on ball.)

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
Offset 10h-13h	**LAD_EN — LPC Address Enable Register (R/W)** **Reset Value: 00000000h**
31:18	**Reserved.**
17	**LPC RTC.** RTC addresses I/O Ports 070h-073h. See bit 16 for decode.
16	**LPC/ISA Default Mapping.** Works in conjunction with bits 17 and [14:0] of this register to enable mapping of specific peripherals to LPC or internal ISA interfaces. If bit [x] = 0 and bit 16 = 0 then: Transaction routed to internal ISA bus. If bit [x] = 0 and bit 16 = 1 then: Transaction routed to LPC interface. If bit [x] = 1 and bit 16 = 0 then: Transaction routed to LPC interface. If bit [x] = 1 and bit 16 = 1 then: Transaction routed to internal ISA bus. Bit [x] is defined as bits 17 and [14:0].
15	**LPC ROM Addressing.** Depends upon F0 Index 52h[2,0]. 0: Disable. 1: Enable.
14	**LPC Alternate SuperI/O Addressing.** Alternate SuperI/O control addresses 4Eh-4Fh. See bit 16 for decode.
13	**LPC SuperI/O Addressing.** SuperI/O control addresses I/O Ports 2Eh-2Fh. See bit 16 for decode. **Note:** This bit should not be enabled when using the internal SuperI/O module and if IO_SIOCFG_IN (F5BAR0+I/O Offset 00h[26:25]) = 11.
12	**LPC Ad-Lib Addressing.** Ad-Lib addresses I/O Ports 388h-389h. See bit 16 for decode.
11	**LPC ACPI Addressing.** ACPI microcontroller addresses I/O Ports 62h and 66h. See bit 16 for decode.
10	**LPC Keyboard Controller Addressing.** KBC addresses I/O Ports 60h and 64h. **Note:** If this bit = 0 and bit 16 = 1, then F0 Index 5Ah[1] must be written 0.
9	**LPC Wide Generic Addressing.** Wide generic addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 18h[15:9]. **Note:** The selected range must not overlap any address range that is positively decoded by F0BAR1+I/O Offset 10h bits [17], [14:10], and [8:0].
8	**LPC Game Port 1 Addressing.** Game Port 1 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[22:19].
7	**LPC Game Port 0 Addressing.** Game Port 0 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[18:15].
6	**LPC Floppy Disk Controller Addressing.** FDC addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[14].
5	**LPC Microsoft Sound System (MSS) Addressing.** MSS addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[13:12].
4	**LPC MIDI Addressing.** MIDI addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[11:10].
3	**LPC Audio Addressing.** Audio addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[9:8].
2	**LPC Serial Port 1 Addressing.** Serial Port 1 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[7:5].
1	**LPC Serial Port 0 Addressing.** Serial Port 0 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[4:2].
0	**LPC Parallel Port Addressing.** Parallel Port addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[1:0].

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
Offset 14h-17h	**LAD_D0 — LPC Address Decode 0 Register (R/W)** — **Reset Value: 00080020h**
31:23	**Reserved.**
22:19	**LPC Game Port 1 Address Select.** Selects I/O Port: 0000: 200h 0100: 204h 1000: 208h 1100: 20Ch 0001: 201h 0101: 205h 1001: 209h 1101: 20Dh 0010: 202h 0110: 206h 1010: 20Ah 1110: 20Eh 0011: 203h 0111: 207h 1011: 20Bh 1111: 20Fh Selected address range is enabled via F0BAR1+I/O Offset 10h[8].
18:15	**LPC Game Port 0 Address Select.** Selects I/O Port: 0000: 200h 0100: 204h 1000: 208h 1100: 20Ch 0001: 201h 0101: 205h 1001: 209h 1101: 20Dh 0010: 202h 0110: 206h 1010: 20Ah 1110: 20Eh 0011: 203h 0111: 207h 1011: 20Bh 1111: 20Fh Selected address range is enabled via F0BAR1+I/O Offset 10h[7].
14	**LPC Floppy Disk Controller Address Select.** Selects I/O Port: 0: 3F0h-3F7h. 1: 370h-377h. Selected address range is enabled via F0BAR1+I/O Offset 10h[6].
13:12	**LPC Microsoft Sound System (MSS) Address Select.** Selects I/O Port: 00: 530h-537h 10: E80h-E87h 01: 604h-60Bh 11: F40h-F47h Selected address range is enabled via F0BAR1+I/O Offset 10h[5].
11:10	**LPC MIDI Address Select.** Selects I/O Port: 00: 300h-301h 10: 320h-321h 01: 310h-311h 11: 330h-331h Selected address range is enabled via F0BAR1+I/O Offset 10h[4].
9:8	**LPC Audio Address Select.** Selects I/O Port: 00: 220h-233h 10: 260h-273h 01: 240h-253h 11: 280h-293h Selected address range is enabled via F0BAR1+I/O Offset 10h[3].
7:5	**LPC Serial Port 1 Address Select.** Selects I/O Port: 000: 3F8h-3FFh 010: 220h-227h 100: 238h-23Fh 110: 338h-33Fh 001: 2F8h-2FFh 011: 228h-22Fh 101: 2E8h-2EFh 111: 3E8h-3EFh Selected address range is enabled via F0BAR1+I/O Offset 10h[2].
4:2	**LPC Serial Port 0 Address Select.** Selects I/O Port: 000: 3F8h-3FFh 010: 220h-227h 100: 238h-23Fh 110: 338h-33Fh 001: 2F8h-2FFh 011: 228h-22Fh 101: 2E8h-2EFh 111: 3E8h-3EFh Selected address range is enabled via F0BAR1+I/O Offset 10h[1].
1:0	**LPC Parallel Port Address Select.** Selects I/O Port: 00: 378h-37Fh (+778h-77Fh for ECP) 01: 278h-27Fh (+678h-67Fh for ECP) (Note) 10: 3BCh-3BFh (+7BCh-7BFh for ECP) 11: Reserved Selected address range is enabled via F0BAR1+I/O Offset 10h[0]. **Note:** 279h is read only, writes are forwarded to ISA for PnP.
Offset 18h-1Bh	**LAD_D1 — LPC Address Decode 1 Register (R/W)** — **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:9	**Wide Generic Base Address Select.** Defines a 512 byte space. Can be mapped anywhere in the 64 KB I/O space. AC97 and other configuration registers are expected to be mapped to this range. It is wide enough to allow many unforeseen devices to be supported. Enabled at F0BAR1+I/O Offset 10h[9]. **Note:** The selected range must not overlap any address range that is positively decoded by F0BAR1+I/O Offset 10h bits [17], [14:10], and [8:0].
8:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
Offset 1Ch-1Fh	**LPC_ERR_SMI — LPC Error SMI Register (R/W)** **Reset Value: 00000080h**
31:12	**Reserved.** Must be set to 0.
11	**LPCPD# Override Enable.** Determines how LPCPD# output is controlled. 0: ACPI logic. 1: LPCPD# Override Value bit (bit 10 of this register).
10	**LPCPD# Override Value.** Selects value of LPCPD# output if bit 11 of this register is set to 1. 0: Power down sequence. 1: Normal power.
9	**SMI Serial IRQ Enable.** Allows serial IRQ to generate an SMI. 0: Disable. 1: Enable. Top Level SMI status is reported at F1BAR0+I/O Offset 02h[3]. Second level status is reported at bit 6 of this register.
8	**SMI Configuration for LPC Error Enable.** Allows LPC errors to generate an SMI. 0: Disable. 1: Enable. Top Level SMI status is reported at F1BAR0+I/O Offset 02h[3]. Second level status is reported at bit 5 of this register.
7	**LPCPD# Pin Status. (Read Only)** Reflects the current value of the LPCPD# output signal.
6	**SMI Source is Serial IRQ.** Indicates whether or not an SMI was generated by an SERIRQ. 0: No. 1: Yes. Write 1 to clear. To enable SMI generation, set bit 9 of this register to 1. This is the second level of status reporting. The top level status is reported in F1BAR0+I/O Offset 02h[3]. Writing a 1 to this bit also clears the top level status bit as long as bit 5 of this register is cleared.
5	**LPC Error Status.** Indicates whether or not an SMI was generated by an error that occurred on LPC. 0: No. 1: Yes. Write 1 to clear. To enable SMI generation, set bit 8 of this register to 1. This is the second level of status reporting. The top level status is reported in F1BAR0+I/O Offset 02h[3]. Writing a 1 to this bit also clears the top level status bit as long as bit 6 of this register is cleared.
4	**LPC Multiple Errors Status.** Indicates whether or not multiple errors have occurred on LPC. 0: No. 1: Yes. Write 1 to clear.
3	**LPC Timeout Error Status.** Indicates whether or not an error was generated by a timeout on LPC. 0: No. 1: Yes. Write 1 to clear.
2	**LPC Error Write Status.** Indicates whether or not an error was generated during a write operation on LPC. 0: No. 1: Yes. Write 1 to clear.
1	**LPC Error DMA Status.** Indicates whether or not an error was generated during a DMA operation on LPC. 0: No. 1: Yes. Write 1 to clear.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset: LPC Interface Configuration Registers (Continued)

Bit	Description
0	**LPC Error Memory Status.** Indicates whether or not an error was generated during a memory operation on LPC. 0: No. 1: Yes. Write 1 to clear.
Offset 20h-23h	**LPC_ERR_ADD — LPC Error Address Register (RO)** **Reset Value: 00000000h**
31:0	**LPC Error Address.**

Core Logic Module (Continued)

5.4.2 SMI Status and ACPI Registers - Function 1

The register space designated as Function 1 (F1) is used to configure the PCI portion of support hardware for the SMI Status and ACPI Support registers. The bit formats for the PCI Header registers are given in Table 5-32.

Located in the PCI Header registers of F1 are two Base Address Registers (F1BARx) used for pointing to the register spaces designated for SMI status and ACPI support, described later in this section.

Table 5-32. F1: PCI Header Registers for SMI Status and ACPI Support

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0501h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:1	**Reserved. (Read Only)**
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled to access I/O offsets through F1BAR0 and F1BAR1 (see F1 Index 10h and 40h).
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 068000h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F1BAR0 (R/W)** — **Reset Value: 00000001h** This register allows access to I/O mapped SMI status related registers. Bits [7:0] are read only (0000 0001), indicating a 256-byte I/O address range. Refer to Table 5-33 on page 244 for bit formats and reset values of the SMI status registers.
31:8	**SMI Status Base Address.**
7:0	**Address Range. (Read Only)**
Index 14h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0501h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**
Index 40h-43h	**Base Address Register 1 - F1BAR1 (R/W)** — **Reset Value: 00000001h** This register allows access to I/O mapped ACPI related registers. Bits [7:0] are read only (0000 0001), indicating a 256 byte address range. Refer to Table 5-34 on page 253 for bit formats and reset values of the ACPI registers. **Note:** This Base Address register moved from its normal PCI Header Space (F1 Index 14h) to prevent plug and play software from relocating it after an FACP table is built.
31:8	**ACPI Base Address.**
7:1	**Address Range. (Read Only)**
0	**Enable. (Write Only)** This bit must be set to 1 to enable access to ACPI Support Registers.
Index 44h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

5.4.2.1 SMI Status Support Registers

F1 Index 10h, Base Address Register 0 (F1BAR0), points to the base address for SMI Status register locations. Table 5-33 gives the bit formats of I/O mapped SMI Status registers accessed through F1BAR0.

The registers at F1BAR0+I/O Offset 50h-FFh can also be accessed F0 Index 50h-FFh. The preferred method is to program these registers through the F0 register space.

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers

Bit	Description
Offset 00h-01h	**Top Level PME/SMI Status Mirror Register (RO)** **Reset Value: 0000h**
Note:	Reading this register does not clear the status bits. For more information, see F1BAR0+I/O Offset 02h.
15	**Suspend Modulation Enable Mirror.** This bit mirrors the Suspend Mode Configuration bit (F0 Index 96h[0]). It is used by the SMI handler to determine if the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h) must be cleared on exit.
14	**SMI Source is USB.** Indicates whether or not an SMI was caused by USB activity. 0: No. 1: Yes. To enable SMI generation, set F5BAR0+I/O Offset 00h[20:19] to 11.
13	**SMI Source is Warm Reset Command.** Indicates whether or not an SMI was caused by a Warm Reset command. 0: No. 1: Yes.
12	**SMI Source is NMI.** Indicates whether or not an SMI was caused by NMI activity. 0: No. 1: Yes.
11	**SMI Source is SuperI/O.** Indicates whether or not an SMI was caused by SuperI/O. 0: No. 1: Yes. The next level (second level) of SMI status is reported in the SuperI/O module. For more information, see Table 4-29 "Banks 0 and 1 - Common Control and Status Registers" on page 127, Offset 00h.
10	**SMI Source is EXT_SMI[7:0].** Indicates whether or not an SMI was caused by a negative-edge event on EXT_SMI[7:0]. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 24h[23:8].
9	**SMI Source is GP Timers/UDEF/PCI/ISA Function Trap.** Indicates if an SMI was caused by: — Expiration of GP Timer 1 or 2. — Trapped access to UDEF1, 2, or 3. — Trapped access to F1-F5 or ISA Legacy register space. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 04h/06h.
8	**SMI Source is Software Generated.** Indicates whether or not an SMI was caused by software. 0: No. 1: Yes.
7	**SMI on an A20M# Toggle.** Indicates whether or not an SMI was caused by a write access to either Port 92h or the keyboard command which initiates an A20M# SMI. 0: No. 1: Yes. This method of controlling the internal A20M# in the GX1 module is used instead of a pin. To enable SMI generation, set F0 Index 53h[0] to 1.
6	**SMI Source is a VGA Timer Event.** Indicates whether or not an SMI was caused by the expiration of the VGA Timer (F0 Index 8Eh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[3] to 1.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
5	**SMI Source is Video Retrace.** Indicates whether or not an SMI was caused by a video retrace event as decoded from the internal serial connection (PSERIAL register, bit 7) from the GX1 module. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[2] to 1.
4	**Reserved.** Reads as 0.
3	**SMI Source is LPC.** Indicates whether or not an SMI was caused by the LPC interface. 0: No. 1: Yes. The next level (second level) of SMI status is at F0BAR1+I/O Offset 1Ch[6:5].
2	**SMI Source is ACPI.** Indicates whether or not an SMI was caused by an access (read or write) to one of the ACPI registers (F1BAR1). 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 20h.
1	**SMI Source is Audio Subsystem.** Indicates whether or not an SMI was caused by the audio subsystem. 0: No. 1: Yes. The next level (second level) of SMI status is at F3BAR0+Memory Offset 10h/12h.
0	**SMI Source is Power Management Event.** Indicates whether or not an SMI was caused by one of the power management resources (except for GP timers, UDEFx and PCI/ISA function traps that are reported in bit 9). 0: No. 1: Yes. The next level (second level) of SMI status is at F0 Index 84h-F4h/87h-F7h.

Offset 02h-03h **Top Level PME/SMI Status Register (RO/RC)** **Reset Value: 0000h**

Note: Reading this register clears all the SMI status bits except for the “read only” bits, because they have a second level of status reporting. Clearing the second level status bits also clears the top level (except for GPIOs).

GPIO SMIs have third level of SMI status reporting at F0BAR0+I/O Offset 0Ch/1Ch. Clearing the third level GPIO status bits also clears the second and top levels.

A read-only “Mirror” version of this register exists at F1BAR0+I/O Offset 00h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F1BAR0+I/O Offset 00h can be read instead.

15	**Suspend Modulation Enable Mirror. (Read to Clear)** This bit mirrors the Suspend Mode Configuration bit (F0 Index 96h[0]). It is used by the SMI handler to determine if the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h) must be cleared on exit.
14	**SMI Source is USB. (Read to Clear)** Indicates whether or not an SMI was caused by USB activity. 0: No. 1: Yes. To enable SMI generation, set F5BAR0+I/O Offset 00h[20:19] to 11.
13	**SMI Source is Warm Reset Command. (Read to Clear)** Indicates whether or not an SMI was caused by Warm Reset command. 0: No. 1: Yes.
12	**SMI Source is NMI. (Read to Clear)** Indicates whether or not an SMI was caused by NMI activity. 0: No. 1: Yes.
11	**SMI Source is SuperI/O. (Read to Clear)** Indicates whether or not an SMI was caused by SuperI/O. 0: No. 1: Yes. The next level (second level) of SMI status is reported in the SuperI/O module. See Table 4-29 "Banks 0 and 1 - Common Control and Status Registers" on page 127 for details.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
10	**SMI Source is EXT_SMI[7:0]. (Read Only. Read Does Not Clear)** Indicates whether or not an SMI was caused by a negative-edge event on EXT_SMI[7:0]. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 24h[23:8].
9	**SMI Source is General Timers/Traps. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by the expiration of one of the General Purpose Timers or one of the User Defined Traps. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 04h/06h.
8	**SMI Source is Software Generated. (Read to Clear)** Indicates whether or not an SMI was caused by software. 0: No. 1: Yes.
7	**SMI on an A20M# Toggle. (Read to Clear)** Indicates whether or not an SMI was caused by an access to either Port 92h or the keyboard command which initiates an A20M# SMI. 0: No. 1: Yes. This method of controlling the internal A20M# in the GX1 module is used instead of a pin. To enable SMI generation, set F0 Index 53h[0] to 1.
6	**SMI Source is a VGA Timer Event. (Read to Clear)** Indicates whether or not an SMI was caused by expiration of the VGA Timer (F0 Index 8Eh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[3] to 1.
5	**SMI Source is Video Retrace. (Read to Clear)** Indicates whether or not an SMI was caused by a video retrace event as decoded from the internal serial connection (PSERIAL register, bit 7) from the GX1 module. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[2] to 1.
4	**Reserved.** Reads as 0.
3	**SMI Source is LPC. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by the LPC interface. 0: No. 1: Yes. The next level (second level) of SMI status is at F0BAR1+I/O Offset 1Ch[6:5].
2	**SMI Source is ACPI. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by an access (read or write) to one of the ACPI registers (F1BAR1). 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 20h.
1	**SMI Source is Audio Subsystem. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by the audio subsystem. 0: No. 1: Yes. The second level of status is found in F3BAR0+Memory Offset 10h/12h.
0	**SMI Source is Power Management Event. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by one of the power management resources (except for GP timers, UDEFx and PCI/ISA function traps which are reported in bit 9). 0: No. 1: Yes. The next level (second level) of SMI status is at F0 Index 84h/F4h-87h/F7h.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
Offset 04h-05h	**Second Level General Traps & Timers PME/SMI Status Mirror Register (RO)** — **Reset Value: 0000h**
The bits in this register contain second level status reporting. Top level status is reported at F1BAR0+I/O Offset 00h/02h[9]. Reading this register does not clear the SMI. For more information, see F1BAR0+I/O Offset 06h.	
15:6	**Reserved.**
5	**PCI/ISA Function Trap.** Indicates whether or not an SMI was caused by a trapped PCI/ISA configuration cycle. 0: No. 1: Yes. To enable SMI generation for: — Trapped access to ISA Legacy I/O register space set F0 Index 41h[0] = 1. — Trapped access to F1 register space set F0 Index 41h[1] = 1. — Trapped access to F2 register space set F0 Index 41h[2] = 1. — Trapped access to F3 register space set F0 Index 41h[3] = 1. — Trapped access to F4 register space set F0 Index 41h[4] = 1. — Trapped access to F5 register space set F0 Index 41h[5] = 1.
4	**SMI Source is Trapped Access to User Defined Device 3.** Indicates whether or not an SMI was caused by a trapped I/O or memory access to the User Defined Device 3 (F0 Index C8h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[6] = 1.
3	**SMI Source is Trapped Access to User Defined Device 2.** Indicates whether or not an SMI was caused by a trapped I/O or memory access to the User Defined Device 2 (F0 Index C4h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[5] = 1.
2	**SMI Source is Trapped Access to User Defined Device 1.** Indicates whether or not an SMI was caused by a trapped I/O or memory access to the User Defined Device 1 (F0 Index C0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[4] = 1.
1	**SMI Source is Expired General Purpose Timer 2.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 2 (F0 Index 8Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[1] = 1.
0	**SMI Source is Expired General Purpose Timer 1.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 1 (F0 Index 88h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[0] = 1.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description	
Offset 06h-07h	**Second Level General Traps & Timers Status Register (RC)**	**Reset Value: 0000h**
The bits in this register contain second level of status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[9]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F1BAR0+I/O Offset 04h. If the value of this register must be read without clearing the SMI source (and consequently de-asserting SMI), F1BAR0+I/O Offset 04h can be read instead.		
15:6	**Reserved.**	
5	**PCI/ISA Function Trap.** Indicates whether or not an SMI was caused by a trapped PCI/ISA configuration cycle 0: No.1:Yes. To enable SMI generation for: — Trapped access to ISA Legacy I/O register space set F0 Index 41h[0] = 1. — Trapped access to F1 register space set F0 Index 41h[1] = 1. — Trapped access to F2 register space set F0 Index 41h[2] = 1. — Trapped access to F3 register space set F0 Index 41h[3] = 1. — Trapped access to F4 register space set F0 Index 41h[4] = 1. — Trapped access to F5 register space set F0 Index 41h[5] = 1.	
4	**SMI Source is Trapped Access to User Defined Device 3 (UDEF3).** Indicates whether or not an SMI was caused by a trapped I/O or memory access to User Defined Device 3 (F0 Index C8h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[6] = 1.	
3	**SMI Source is Trapped Access to User Defined Device 2 (UDEF2).** Indicates whether or not an SMI was caused by a trapped I/O or memory access to User Defined Device 2 (F0 Index C4h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[5] = 1.	
2	**SMI Source is Trapped Access to User Defined Device 1 (UDEF1).** Indicates whether or not an SMI was caused by a trapped I/O or memory access to User Defined Device 1 (F0 Index C0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[4] = 1.	
1	**SMI Source is Expired General Purpose Timer 2.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 2 (F0 Index 8Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[1] = 1.	
0	**SMI Source is Expired General Purpose Timer 1.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 1 (F0 Index 88h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[0] = 1.	
Offset 08h-09h	**SMI Speedup Disable Register (Read to Enable)**	**Reset Value: 0000h**
15:0	**SMI Speedup Disable.** If bit 1 in the Suspend Configuration Register is set (F0 Index 96h[1] = 1), a read of this register invokes the SMI handler to re-enable Suspend Modulation. The data read from this register can be ignored. If the Suspend Modulation feature is disabled, reading this I/O location has no effect.	
Offset 0Ah-1Bh	**Reserved**	**Reset Value: 00h**
These addresses should not be written.		
Offset 1Ch-1Fh	**ACPI Timer Register (RO)**	**Reset Value: xxxxxxxxh**
Note: This register can also be read at F1BAR1+I/O Offset 1Ch.		
31:24	**Reserved.**	
23:0	**TMR_VAL.** This field returns the running count of the power management timer.	

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
Offset 20h-21h	**Second Level ACPI PME/SMI Status Mirror Register (RO)** — **Reset Value: 0000h**
The bits in this register contain second level SMI status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[2]. Reading this register does not clear the SMI. For more information, see F1BAR0+I/O Offset 22h.	
15:6	**Reserved.** Always reads 0.
5	**ACPI BIOS SMI Status.** Indicates whether or not an SMI was caused by ACPI software raising an event to BIOS software. 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[2] to 1, and F1BAR1+I/O Offset 0Fh[0] to 1.
4	**PLVL3 SMI Status.** Indicates whether or not an SMI was caused by a read of the ACPI PLVL3 register (F1BAR1+I/O Offset 05h). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[11] to 1 (default).
3	**Reserved.**
2	**SLP_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[9] to 1 (default).
1	**THT_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI THT_EN bit (F1BAR1+I/O Offset 00h[4]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[8] to 1 (default).
0	**SMI_CMD SMI Status.** Indicates whether or not an SMI was caused by a write to the ACPI SMI_CMD register (F1BAR1+I/O Offset 06h). 0: No. 1: Yes. A write to the ACPI SMI_CMD register always generates an SMI.
Offset 22h-23h	**Second Level ACPI PME/SMI Status Register (RC)** — **Reset Value: 0000h**
The bits in this register contain second level of SMI status reporting. Top level is reported in F1BAR0+I/O Offset 00h/02h[2]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F1BAR0+I/O Offset 20h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F1BAR0+I/O Offset 20h can be read instead.	
15:6	**Reserved.** Always reads 0.
5	**ACPI BIOS SMI Status.** Indicates whether or not an SMI was caused by ACPI software raising an event to BIOS software. 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[2] to 1, and F1BAR1+I/O Offset 0Fh[0] to 1.
4	**PLVL3 SMI Status.** Indicates whether or not an SMI was caused by a read of the ACPI PLVL3 register (F1BAR1+I/O Offset 05h). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[11] to 1 (default).
3	**Reserved.**
2	**SLP_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[9] to 1 (default).

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
1	**THT_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI THT_EN bit (F1BAR1+I/O Offset 00h[4]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[8] to 1 (default).
0	**SMI_CMD SMI Status.** Indicates whether or not an SMI was caused by a write to the ACPI SMI_CMD register (F1BAR1+I/O Offset 06h). 0: No. 1: Yes. A write to the ACPI SMI_CMD register always generates an SMI.

Offset 24h-27h **External SMI Register (R/W)** **Reset Value: 00000000h**

Note: EXT_SMI[7:0] are external SMIs, meaning external to the Core Logic module.

Bits [23:8] of this register contain second level of SMI status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[10]. Reading bits [23:16] clears the second and top levels. If the value of the status bits must be read without clearing the SMI source (and consequently de-asserting SMI), bits [15:8] can be read instead.

31:24	**Reserved.** Must be set to 0.
23	**EXT_SMI7 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by assertion of EXT_SMI7. 0: No. 1: Yes. To enable SMI generation, set bit 7 to 1.
22	**EXT_SMI6 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI6. 0: No. 1: Yes. To enable SMI generation, set bit 6 to 1.
21	**EXT_SMI5 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI5. 0: No. 1: Yes. To enable SMI generation, set bit 5 to 1.
20	**EXT_SMI4 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI4. 0: No. 1: Yes. To enable SMI generation, set bit 4 to 1.
19	**EXT_SMI3 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI3. 0: No. 1: Yes. To enable SMI generation, set bit 3 to 1.
18	**EXT_SMI2 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI2. 0: No. 1: Yes. To enable SMI generation, set bit 2 to 1.
17	**EXT_SMI1 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI1. 0: No. 1: Yes. To enable SMI generation, set bit 1 to 1.
16	**EXT_SMI0 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI0. 0: No. 1: Yes. To enable SMI generation, set bit 0 to 1.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
15	**EXT_SMI7 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI7. 0: No. 1: Yes. To enable SMI generation, set bit 7 to 1.
14	**EXT_SMI6 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI6. 0: No. 1: Yes. To enable SMI generation, set bit 6 to 1.
13	**EXT_SMI5 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI5. 0: No. 1: Yes. To enable SMI generation, set bit 5 to 1.
12	**EXT_SMI4 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI4. 0: No. 1: Yes. To enable SMI generation, set bit 4 to 1.
11	**EXT_SMI3 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI3. 0: No. 1: Yes. To enable SMI generation, set bit 3 to 1.
10	**EXT_SMI2 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI2. 0: No. 1: Yes. To enable SMI generation, set bit 2 to 1.
9	**EXT_SMI1 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI1. 0: No. 1: Yes. To enable SMI generation, set bit 1 to 1.
8	**EXT_SMI0 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI0. 0: No. 1: Yes. To enable SMI generation, set bit 0 to 1.
7	**EXT_SMI7 SMI Enable.** When this bit is asserted, allow EXT_SMI7 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 23 (RC) and 15 (RO).
6	**EXT_SMI6 SMI Enable.** When this bit is asserted, allow EXT_SMI6 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 22 (RC) and 14 (RO).
5	**EXT_SMI5 SMI Enable.** When this bit is asserted, allow EXT_SMI5 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 21 (RC) and 13 (RO).

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
4	**EXT_SMI4 SMI Enable.** When this bit is asserted, allows EXT_SMI4 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 20 (RC) and 12 (RO).
3	**EXT_SMI3 SMI Enable.** When this bit is asserted, allow EXT_SMI3 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 19 (RC) and 11 (RO).
2	**EXT_SMI2 SMI Enable.** When this bit is asserted, allow EXT_SMI2 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 18 (RC) and 10 (RO).
1	**EXT_SMI1 SMI Enable.** When this bit is asserted, allow EXT_SMI1 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 17 (RC) and 9 (RO).
0	**EXT_SMI0 SMI Enable.** When this bit is asserted, allow EXT_SMI0 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 16 (RC) and 8 (RO).
Offset 28h-4Fh	**Not Used** **Reset Value: 00h**
Offset 50h-FFh	The I/O mapped registers located here (F1BAR0+I/O Offset 50h-FFh) can also be accessed at F0 Index 50h-FFh. The preferred method is to program these registers through the F0 register space. Refer to Table 5-29 "F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support" on page 199 for more information about these registers.

5.4.2.2 ACPI Support Registers

F1 Index 40h, Base Address Register 1 (F1BAR1), points to the base address of where the ACPI Support registers are located. Table 5-34 shows the I/O mapped ACPI Support registers accessed through F1BAR1.

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers

Bit	Description
Offset 00h-03h	**P_CNT — Processor Control Register (R/W)** — **Reset Value: 00000000h**
31:5	**Reserved.** Always reads 0.
4	**THT_EN (Throttle Enable).** When this bit is asserted, it enables throttling of the clock based on the CLK_VAL field (bits [2:0] of this register). 0: Disable. 1: Enable. If F1BAR1+I/O Offset 18h[8] =1, an SMI is generated when this bit is set. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[1].
3	**Reserved.** Always reads 0.
2:0	**CLK_VAL (Clock Throttling Value).** CPU duty cycle: 000: Reserved — 010: 25% — 100: 50% — 110: 75% 001: 12.5% — 011: 37.5% — 101: 62.5% — 111: 87.5%
Offset 04h	**Reserved** — **Reset Value: 00h**
Note:	**This register should not be read.** It controls a reserved function of power management logic.
Offset 05h	**P_LVL3 — Enter C3 Power State Register (RO)** — **Reset Value: xxh**
7:0	**P_LVL3 (Power Level 3).** Reading this 8-bit read only register causes the processor to enter the C3 power state. Reads of P_LVL3 return 0. Writes have no effect. The ACPI state machine always waits for an SMI (any SMI) to be generated and serviced before transfer into C3 power state. A read of this register causes an SMI if enabled: F1BAR1+I/O Offset 18h[11] = 1 (default). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[4].
Offset 06h	**SMI_CMD — OS/BIOS Requests Register (R/W)** — **Reset Value: 00h**
7:0	**SMI_CMD (SMI Command and OS / BIOS Requests).** A write to this register stores data and a read returns the last data written. In addition, a write to this register always generates an SMI. A read of this register does not generate an SMI. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[0].
Offset 07h	**ACPI_FUN_CNT — ACPI Function Control Register (R/W)** — **Reset Value: 00h**
7:6	**LED_CNT (LED Output Control).** Controls the blinking of an LED when in the SL4 or SL5 sleep state. 00: Disable (LED# signal, is HIZ). 01: Zero (LED# signal is HIZ). 10: Blink @ 1 Hz rate, when in SL4 and SL5 sleep states. Duty cycle: LED# is 10% pulled low, 90% HIZ. 11: One (LED# is pulled low, when in SL4 and SL5 sleep states).
5	**Reserved.** Must be set to 0.
4	**INTR_WU_SL1.** Enables wakeup on enabled interrupts in sleep state SL1. 0: Disable wakeup from SL1, when an enabled interrupt is active. 1: Enable wakeup from SL1, when an enabled interrupt is active.
3	**GPWIO_DBNC_DIS (GPWIO0 and GPWIO1 Debouncers).** Debounce settings for GPWIO0 and GPWIO1. Selects the time that a high-to-low or low-to-high transition (debounce period) must be for GPWIO0 to be recognized. 0: Debounce period is 15.8 msec. (Default) 1: Debounce period is 31 µs. GPWIO2 pin is fixed at 31 µs.
2:1	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
0	**PWRBTN_DBNC_DIS (Power Button Debouncer).** Allow a high-to-low or low-to-high transition of greater than 15.8 ms (debounce period) on PWRBTN# before it is recognized. 0: Enable. (Default) 1: Disable.
Offset 08h-09h	**PM1A_STS — PM1A Top Level PME/SCI Status Register (R/W)** **Reset Value: 0000h**
Notes: 1. This is the top level of PME/SCI status reporting for these events. There is no second level. 2. If SCI generation is not desired, the status bits are still set by the described conditions and can be used for monitoring purposes.	
15	**WAK_STS (Wakeup Status).** Indicates whether or not an SCI was caused by the occurrence of an enabled wakeup event. 0: No. 1: Yes. This bit is set when the system is in any Sleep state and an enabled wakeup event occurs (wakeup events are configured at F1BAR1+I/O Offset 0Ah and 12h). After this bit is set, the system transitions to a Working state. SCI generation is always enabled. Write 1 to clear.
14:12	**Reserved.** Must be set to 0.
11	**PWRBTNOR_STS (Power Button Override Status).** Indicates whether or not an SCI was caused by the power button being active for greater than 4 seconds. 0: No. 1: Yes. SCI generation is always enabled. Write 1 to clear.
10	**RTC_STS (Real-Time Clock Status).** Indicates if a Power Management Event (PME) was caused by the RTC generating an alarm (RTC IRQ signal is asserted). 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[10] to 1 and F1BAR1+I/O Offset 0Ch[0] to 1. (See Note 2 in the general description of this register.) Write 1 to clear.
9	**Reserved.** Must be set to 0.
8	**PWRBTN_STS (Power Button Status).** Indicates if PME was caused by the PWRBTN# going low while the system is in a Working state. 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[8] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register.) In a Sleep state or the Soft-Off state, a wakeup event is generated when the power button is pressed (regardless of the PWRBTN_EN bit, F1BAR1+I/O Offset 0Ah[8], setting). Write 1 to clear.
7:6	**Reserved.** Must be set to 0.
5	**GBL_STS (Global Lock Status).** Indicates if PME was caused by the BIOS releasing control of the global lock. 0: No. 1: Yes. This bit is used by the BIOS to generate an SCI. BIOS writes the BIOS_RLS bit (F1BAR1+I/O Offset 0Fh[1]) which in turns sets the GBL_STS bit and raises a PME. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[5] to 1 and F1BAR1+I/O Offset 0Ch[0] to 1. (See Note 2 in the general description of this register.) Write 1 to clear.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
4	**BM_STS (Bus Master Status).** Indicates if PME was caused by a system bus master requesting the system bus. 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ch[1] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register.) Write 1 to clear.
3:1	**Reserved.** Must be set to 0.
0	**TMR_STS (Timer Carry Status).** Indicates if SCI was caused by an MSB toggle (MSB changes from low-to-high or high-to-low) on the ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch). 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[0] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register.) Write 1 to clear.

Offset 0Ah-0Bh **PM1A_EN — PM1A PME/SCI Enable Register (R/W)** **Reset Value: 0000h**

In order for the ACPI events described below to generate an SCI, the SCI_EN bit must also be set (F1BAR1+I/O Offset 0Ch[0] = 1).

The SCIs enabled via this register are globally enabled by setting F1BAR1+I/O Offset 08h. There is no second level of SCI status reporting for these bits.

15:11	**Reserved.** Must be set to 0.
10	**RTC_EN (Real-Time Clock Enable).** Allow SCI generation when the RTC generates an alarm (RTC IRQ signal is asserted). 0: Disable. 1: Enable
9	**Reserved.** Must be set to 0.
8	**PWRBTN_EN (Power Button Enable).** Allow SCI generation when PWRBTN# goes low while the system is in a Working state. 0: Disable. 1: Enable.
7:6	**Reserved.** Must be set to 0.
5	**GBL_EN (Global Lock Enable).** Allow SCI generation when the BIOS releases control of the global lock via the BIOS_RLS (F1BAR1+I/O Offset 0Fh[1] and GBL_STS (F1BAR1+I/O Offset 08h[5]) bits. 0: Disable. 1: Enable.
4:1	**Reserved.** Must be set to 0.
0	**TMR_EN (ACPI Timer Enable).** Allow SCI generation for MSB toggles (MSB changes from low-to-high or high-to-low) on the ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch). 0: Disable. 1: Enable.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
Offset 0Ch-0Dh	**PM1A_CNT — PM1A Control Register (R/W)** **Reset Value: 0000h**
15:14	**Reserved.** Must be set to 0.
13	**SLP_EN (Sleep Enable). (Write Only)** Allow the system to sequence into the sleeping state associated with the SLP_TYPx (bits [12:10]). 0: Disable. 1: Enable. This is a write only bit and reads of this bit always return a 0. The ACPI state machine always waits for an SMI (any SMI) to be generated and serviced before transitioning into a Sleep state. If F1BAR1+I/O Offset 18h[9] = 1, an SMI is generated when SLP_EN is set. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[2].
12:10	**SLP_TYPx (Sleep Type).** Defines the type of Sleep state the system enters when SLP_EN (bit 13) is set. 000: Sleep State S0 (Full on) 100: Sleep State SL4 001: Sleep State SL1 101: Sleep State SL5 (Soft off) 010: Sleep State SL2 110: Reserved 011: Sleep State SL3 111: Reserved
9:3	**Reserved.** Set to 0.
2	**GBL_RLS (Global Release). (Write Only)** This write only bit is used by ACPI software to raise an event to the BIOS software (i.e., it generates an SMI to pass execution control to the BIOS). 0: Disable. 1: Enable. This is a write only bit and reads of this bit always return a 0. To generate an SMI, ACPI software writes the GBL_RLS bit which in turn sets the BIOS_STS bit (F1BAR1+I/O Offset 0Eh[0]) and raises a PME. For the PME to generate an SMI, set BIOS_EN (F1BAR1+I/O Offset 0Fh[0] to 1). The top level SMI status is reported at F1BAR0+I/O offset 00h/02h. Second level status is at F1BAR0+I/O Offset 22h[5].
1	**BM_RLD (Bus Master RLD).** If the processor is in the C3 state and a bus master request is generated, force the processor to transition to the C0 state. 0: Disable. 1: Enable
0	**SCI_EN (System Control Interrupt Enable).** Globally selects power management events (PMEs) reported in PM1A_STS and GPE0_STS (F1BAR1+I/O Offset 08h and 10h) to be either an SCI or SMI type of interrupt. 0: APM Mode, generates an SMI and status is reported at F1BAR0+I/O Offset 00h/02h[0]. 1: ACPI Mode, generates an SCI if the corresponding PME enable bit is set and status is reported at F1BAR1+I/O Offset 08h and 10h. **Note:** This bit enables the ACPI state machine.
Offset 0Eh	**ACPI_BIOS_STS Register (R/W)** **Reset Value: 00h**
7:1	**Reserved.** Must be set to 0.
0	**BIOS_STS (BIOS Status Release).** When 1 is written to the GLB_RLS bit (F1BAR1+I/O Offset 0Ch[2]), this bit is also set to 1. Write 1 to clear.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
Offset 0Fh	**ACPI_BIOS_EN Register (R/W)** **Reset Value: 00h**
7:2	**Reserved.** Must be set to 0.
1	**BIOS_RLS (BIOS Release). (Write Only)** When this bit is asserted, allow the BIOS to release control of the global lock. 0: Disable. 1: Enable. This is a write only bit and reads of this bit always return a 0. To generate an SCI, the BIOS writes the BIOS_RLS bit which in turn sets the GBL_STS bit (F1BAR1+I/O Offset 08h[5]) and raises a PME. For the PME to generate an SCI, set GBL_EN (F1BAR1+I/O Offset 0Ah[5] to 1).
0	**BIOS_EN (BIOS Enable).** When this bit is asserted, allow SMI generation by ACPI software via writes to GBL_RLS (F1BAR1+I/O Offset 0Ch[2]). 0: Disable. 1: Enable
Offset 10h-11h	**GPE0_STS — General Purpose Event 0 PME/SCI Status Register (R/W)** **Reset Value: xxxxh**
Notes:	1) This is the top level of PME/SCI status reporting. There is no second level except for bit 3 (GPIOs) where the next level of status is reported at F0BAR0+I/O Offset 0Ch/1Ch. 2) If SCI generation is not desired, the status bits are still set by the described conditions and can be used for monitoring purposes.
15:12	**Reserved.** Must be set to 0.
11	**Reserved.**
10	**GPWIO2_STS.** Indicates if PME was caused by activity on GPWIO2. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI: 1) Ensure that GPWIO2 is enabled as an input (F1BAR1+I/O Offset 15h[2] = 0). 2) Set F1BAR1+I/O Offset 12h[10] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.) If F1BAR1+I/O Offset 15h[6] = 1 it overrides these settings and GPWIO2 generates an SMI and the status is reported in F1BAR0+00h/02h[0].
9	**GPWIO1_STS.** Indicates if PME was caused by activity on GPWIO1. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI: 1) Ensure that GPWIO1 is enabled as an input (F1BAR1+I/O Offset 15h[1] = 0) 2) Set F1BAR1+I/O Offset 12h[9] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.) If F1BAR1+I/O Offset 15h[5] = 1 it overrides these settings and GPWIO1 generates an SMI and the status is reported in F1BAR0+00h/02h[0].
8	**GPWIO0_STS.** Indicates if PME was caused by activity on GPWIO0. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI: 1) Ensure that GPWIO0 is enabled as an input (F1BAR1+I/O Offset 15h[0] = 0). 2) Set F1BAR1+I/O Offset 12h[8] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above). If F1BAR1+I/O Offset 15h[4] = 1 it overrides these settings and GPWIO0 generates an SMI and the status is reported in F1BAR0+00h/02h[0].
7	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
6	**USB_STS.** Indicates if PME was caused by a USB interrupt event. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[6] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.)
5	**THRM_STS.** Indicates if PME was caused by activity on THRM#. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[5] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1, (See Note 2 in the general description of this register above.)
4	**SMI_STS.** Indicates if PME was caused by activity on the internal SMI# signal. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[4] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.)
3	**GPIO_STS.** Indicates if PME was caused by activity on any of the GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0). 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[3] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above). F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled to generate a PME. In addition, the selected GPIO must be enabled as an input (F0BAR0+I/O Offset 20h and 24h).
2:1	**Reserved.** Reads as 0.
0	**PWR_U_REQ_STS.** Indicates if PME was caused by a power-up request event from the SuperI/O module. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[0] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.)

Offset 12h-13h GPE0_EN — General Purpose Event 0 Enable Register (R/W) Reset Value: 0000h

In order for the ACPI events described below to generate an SCI, the SCI_EN bit must also be set (F1BAR1+I/O Offset 0Ch[0] = 1).

The SCIs enabled in this register are globally enabled by setting F1BAR1+I/O Offset 0Ch[0] to 1. The status of the SCIs is reported in F1BAR1+I/O Offset 10h.

15:12	**Reserved.**
11	**Reserved.**
10	**GPWIO2_EN.** Allow GPWIO2 to generate an SCI. 0: Disable. 1: Enable. A fixed high-to-low or low-to-high transition (debounce period) of 31 µs exists in order for GPWIO2 to be recognized. The setting of this bit can be overridden via F1BAR1+I/O Offset 15h[6] to force an SMI.
9	**GPWIO1_EN.** Allow GPWIO1 to generate an SCI. 0: Disable. 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. The setting of this bit can be overridden via F1BAR1+I/O Offset 15h[5] to force an SMI.

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
8	**GPWIO0_EN.** Allow GPWIO0 to generate an SCI. 0: Disable. 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. The setting of this bit can be overridden via F1BAR1+I/O Offset 15h[4] to force an SMI.
7	**Reserved.** Must be set to 0
6	**USB_EN.** Allow USB events to generate a SCI. 0: Disable. 1: Enable.
5	**THRM_EN.** Allow THRM# to generate an SCI. 0: Disable. 1: Enable
4	**SMI_EN.** Allow SMI events to generate an SCI. 0: Disable. 1: Enable.
3	**GPIO_EN.** Allow GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0) to generate an SCI. 0: Disable. 1: Enable. F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled for PME generation. This bit (GPIO_EN) globally enables those selected GPIOs for generation of an SCI.
2:1	**Reserved.** Must be set to 0.
0	**PWR_U_REQ_EN.** Allow power-up request events from the SuperI/O module to generate an SCI. 0: Disable. 1: Enable. A power-up request event is defined as any of the following events/activities: Modem, Telephone, Keyboard, Mouse, CEIR (Consumer Electronic Infrared)

Offset 14h	GPWIO Control Register 1 (R/W)	Reset Value: 00h
7:4	**Reserved.** Must be set to 0.	
3	**Reserved.**	
2	**GPWIO2_POL.** Select GPWIO2 polarity. 0: Active high. 1: Active low.	
1	**GPWIO1_POL.** Select GPWIO1 polarity. 0: Active high. 1: Active low.	
0	**GPWIO0_POL.** Select GPWIO0 polarity. 0: Active high. 1: Active low.	

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
Offset 15h	**GPWIO Control Register 2 (R/W)** — **Reset Value: 00h**
7	**Reserved.**
6	**GPWIO_SMIEN2.** Allow GPWIO2 to generate an SMI. 0: Disable. (Default) 1: Enable. A fixed high-to-low or low-to-high transition (debounce period) of 31 µs exists in order for GPWIO2 to be recognized. Bit 2 of this register must be set to 0 (input) for GPWIO2 to be able to generate an SMI. If asserted, this bit overrides the setting of F1BAR1+I/O Offset 12h[10] and its status is reported in F1BAR0+I/O Offset 00h/02h[0].
5	**GPWIO_SMIEN1.** Allow GPWIO1 to generate an SMI. 0: Disable. (Default) 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. Bit 1 of this register must be set to 0 (input) for GPWIO1 to be able to generate an SMI. If asserted, this bit overrides the setting of F1BAR1+I/O Offset 12h[9] and its status is reported in F1BAR0+I/O Offset 00h/02h[0].
4	**GPWIO_SMIEN0.** Allow GPWIO0 to generate an SMI. 0: Disable. (Default) 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. Bit 0 of this register must be set to 0 (input) for GPWIO0 to be able to generate an SMI. If enabled, this bit overrides the setting of F1BAR1+I/O Offset 12h[8] and its status is reported in F1BAR0+I/O Offset 00h/02h[0].
3	**Reserved.**
2	**GPWIO2_DIR.** Selects the direction of GPWIO2. 0: Input. 1: Output.
1	**GPWIO1_DIR.** Selects the direction of GPWIO1. 0: Input. 1: Output.
0	**GPWIO0_DIR.** Selects the direction of the GPWIO0. 0: Input. 1: Output.

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
Offset 16h	**GPWIO Data Register (R/W)** **Reset Value: 00h**
	This register contains the direct values of the GPWIO2-GPWIO0 pins. Write operations are valid only for bits defined as outputs. Reads from this register read the last written value if the pin is an output. The pins are configured as inputs or outputs in F1BAR1+I/O Offset 15h.
7:4	**Reserved.** Must be set to 0.
3	**Reserved.**
2	**GPWIO2_DATA.** Reflects the level of GPWIO2. 0: Low. 1: High. A fixed high-to-low or low-to-high transition (debounce period) of 31 µs exists in order for GPWIO2 to be recognized.
1	**GPWIO1_DATA.** Reflects the level of GPWIO1. 0: Low. 1: High. See F1BAR1+I/O Offset 07h[3] for debounce information.
0	**GPWIO0_DATA.** Reflects the level of GPWIO0. 0: Low. 1: High. See F1BAR1+I/O Offset 07h[3] for debounce information.
Offset 17h	**Reserved** **Reset Value: 00h**
Offset 18h-1Bh	**ACPI SCI_ROUTING Register (R/W)** **Reset Value: 00000F00h**
31:17	**Reserved.**
16	**PCTL_DELAYEN.** Allow staggered delays on the activation and deactivation of the power control pins PWRCNT1, PWRCNT2, and ONCTL# by 2 msec each. 0: Disable. (Default) 1: Enable.
15:12	**Reserved.** Must be set to 0.
11	**PLVL3_SMIEN.** Allow SMI generation when the PLVL3 Register (F1BAR1+I/O Offset 05h) is read. 0: Disable. 1: Enable. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[4].
10	**Reserved.** Must be set to 0.
9	**SLP_SMIEN.** Allow SMI generation when the SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]) is set. 0: Disable. 1: Enable. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[2].
8	**THT_SMIEN.** Allow SMI generation when the THT_EN bit (F1BAR1+I/O Offset 00h[4]) is set. 0: Disable. 1: Enable. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[1].
7:4	**Reserved.** Must be set to 0.
3:0	**SCI_IRQ_ROUTE.** SCI is routed to: 0000: Disable 0100: IRQ4 1000: IRQ8 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: IRQ13 0010: Reserved 0010: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0011: IRQ7 1011: IRQ11 1111: IRQ15 For more details see Section 5.2.6.3 "Programmable Interrupt Controller" on page 164.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
Offset 1Ch-1Fh	**PM_TMR — ACPI Timer Register (RO)** **Reset Value: xxxxxxxxh**
Note:	This register can also be read at F1BAR0+I/O Offset 1Ch.
31:24	**Reserved.**
23:0	**TMR_VAL. (Read Only)** This bit field contains the running count of the power management timer.
Offset 20h	**PM2_CNT — PM2 Control Register (R/W)** **Reset Value: 00h**
7:1	**Reserved.**
0	**Arbiter Disable.** Disables the PCI arbiter when set by the OS. Used during C3 transition. 0: Arbiter not disabled. (Default) 1: Disable arbiter.
Offset 21h-FFh	**Reserved** **Reset Value: 00h**
The read value for these registers is undefined.	

Core Logic Module (Continued)

5.4.3 IDE Controller Registers - Function 2

The register space designated as Function 2 (F2) is used to configure Channels 0 and 1 and the PCI portion of support hardware for the IDE controllers. The bit formats for the PCI Header/Channels 0 and 1 Registers are given in Table 5-35.

Located in the PCI Header Registers of F2 is a Base Address Register (F2BAR4) used for pointing to the register space designated for support of the IDE controllers, described later in this section.

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0502h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:3	**Reserved. (Read Only)**
2	**Bus Master.** Allow the Core Logic module bus mastering capabilities. 0: Disable. 1: Enable. (Default) This bit must be set to 1.
1	**Reserved. (Read Only)**
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled, in order to access I/O offsets through F2BAR4 (for more information see F2 Index 20h).
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 01h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 010180h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F2BAR0 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 14h-17h	**Base Address Register 1 - F2BAR1 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 18h-1Bh	**Base Address Register 2 - F2BAR2 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 1Ch-1Fh	**Base Address Register 3 - F2BAR3 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 20h-23h	**Base Address Register 4 - F2BAR4 (R/W)** — **Reset Value: 00000001h** **Base Address 0 Register.** This register allows access to I/O mapped Bus Mastering IDE registers. Bits [3:0] are read only (0001), indicating a 16-byte I/O address range. Refer to Table 5-36 on page 267 for the IDE controller register bit formats and reset values.
31:4	**Bus Mastering IDE Base Address.**
3:0	**Address Range. (Read Only)**
Index 24h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0502h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration (Continued)

Bit	Description
Index 40h-43h	**Channel 0 Drive 0 PIO Register (R/W)** — **Reset Value: 00009172h**
If Index 44h[31] = 0, Format 0. The bits in this register select the slowest PIO mode per channel for commands. Format 0 settings for a Fast-PCI clock frequency of 33.3 MHz: — PIO Mode 0 = 00009172h — PIO Mode 1 = 00012171h — PIO Mode 2 = 00020080h — PIO Mode 3 = 00032010h — PIO Mode 4 = 00040010h Format 0 settings for a Fast-PCI clock frequency of 48 MHz: — PIO Mode 0 = 0000FAA3h — PIO Mode 1 = 0001C232h — PIO Mode 2 = 00025131h — PIO Mode 3 = 00033121h — PIO Mode 4 = 00041021h Format 0 settings for a Fast-PCI clock frequency of 66.7 MHz: — PIO Mode 0 = 0000FFF4h — PIO Mode 1 = 0001F353h — PIO Mode 2 = 00028141h — PIO Mode 3 = 00034231h — PIO Mode 4 = 00041131h **Note:** All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.	
31:20	**Reserved.** Must be set to 0.
19:16	**PIOMODE.** PIO mode.
15:12	**t2I.** Recovery time (value + 1 cycle).
11:8	**t3.** IDE_IOW# data setup time (value + 1 cycle).
7:4	**t2W.** IDE_IOW# width minus t3 (value + 1 cycle).
3:0	**t1.** Address Setup Time (value + 1 cycle).
If Index 44h[31] = 1, Format 1. The bits in this register allow independent control of command and data. Format 1 settings for a Fast-PCI clock frequency of 33.3 MHz: — PIO Mode 0 = 9172D132h — PIO Mode 1 = 21717121h — PIO Mode 2 = 00803020h — PIO Mode 3 = 20102010h — PIO Mode 4 = 00100010h Format 1 settings for a Fast-PCI clock frequency of 48 MHz: — PIO Mode 0 = E2A3F383h — PIO Mode 1 = 42A2B232h — PIO Mode 2 = 11B16121h — PIO Mode 3 = 31213121h — PIO Mode 4 = 10211021h Format 1 settings for a Fast-PCI clock frequency of 66.7 MHz: — PIO Mode 0 = F8E4F8E4h — PIO Mode 1 = 53F3F353h — PIO Mode 2 = 13F18141h — PIO Mode 3 = 42314231h — PIO Mode 4 = 11311131h **Note:** All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.	
31:28	**t2IC.** Command cycle recovery time (value + 1 cycle).
27:24	**t3C.** Command cycle IDE_IOW# data setup (value + 1 cycle).
23:20	**t2WC.** Command cycle IDE_IOW# pulse width minus t3 (value + 1 cycle).
19:16	**t1C.** Command cycle address setup time (value + 1 cycle).
15:12	**t2ID.** Data cycle recovery time (value + 1 cycle).
11:8	**t3D.** Data cycle IDE_IOW# data setup (value + 1 cycle).
7:4	**t2WD.** Data cycle IDE_IOW# pulse width minus t3 (value + 1 cycle).
3:0	**t1D.** Data cycle address Setup Time (value + 1 cycle).

Core Logic Module (Continued)

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration (Continued)

Bit	Description
Index 44h-47h	**Channel 0 Drive 0 DMA Control Register (R/W)** — **Reset Value: 00077771h**
The structure of this register depends on the value of bit 20.	
If bit 20 = 0, Multiword DMA	
Settings for a Fast-PCI clock frequency of 33.3 MHz: — Multiword DMA Mode 0 = 00077771h — Multiword DMA Mode 1 = 00012121h — Multiword DMA Mode 2 = 00002020h Settings for a Fast-PCI clock frequency of 48 MHz: — Multiword DMA Mode 0 = 000BBBB2h — Multiword DMA Mode 1 = 00024241h — Multiword DMA Mode 2 = 00013131h Settings for a Fast-PCI clock frequency of 66.7 MHz: — Multiword DMA Mode 0 = 000FFFF3h — Multiword DMA Mode 1 = 00035352h — Multiword DMA Mode 2 = 00015151h	
Note:	All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.
31	**PIO Mode Format.** This bit sets the PIO mode format for all channels and drives. Bit 31 of Offsets 2Ch, 34h, and 3Ch are R/W, but have no function so are defined as reserved. 0: Format 0. 1 Format 1.
30:21	**Reserved.** Must be set to 0.
20	**DMA Select.** Selects type of DMA operation. 0: Multiword DMA
19:16	**tKR.** IDE_IOR# recovery time (4-bit) (value + 1 cycle).
15:12	**tDR.** IDE_IOR# pulse width (value + 1 cycle).
11:8	**tKW**. IDE_IOW# recovery time (4-bit) (value + 1 cycle).
7:4	**tDW**. IDE_IOW# pulse width (value + 1 cycle).
3:0	**tM.** IDE_CS[1:0]# to IDE_IOR#/IOW# setup; IDE_CS[1:0]# setup to IDE_DACK0#/DACK1#.
If bit 20 = 1, UltraDMA	
Settings for a Fast-PCI clock frequency of 33.3 MHz: — UltraDMA Mode 0 = 00921250h — UltraDMA Mode 1 = 00911140h — UltraDMA Mode 2 = 00911030h Settings for a Fast-PCI clock frequency of 48 MHz: — UltraDMA Mode 0 = 00932470h — UltraDMA Mode 1 = 00922260h — UltraDMA Mode 2 = 00922140h Settings for a Fast-PCI clock frequency of 66.7 MHz: — UltraDMA Mode 0 = 009436A1h — UltraDMA Mode 1 = 00933481h — UltraDMA Mode 2 = 00923261h	
Note:	All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.
31	**PIO Mode Format.** This bit sets the PIO mode format for all channels and drives. Bit 31 of Offsets 2Ch, 34h, and 3Ch are R/W, but have no function so are defined as reserved. 0: Format 0. 1: Format 1.
30:24	**Reserved.** Must be set to 0.
23:21	**BSIZE.** Input buffer threshold.
20	**DMA Select.** Selects type of DMA operation. 1: UltraDMA.
19:16	**tCRC.** CRC setup UDMA in IDE_DACK# (value + 1 cycle) (for host terminate CRC setup = tMLI + tSS).
15:12	**tSS.** UDMA out (value + 1 cycle).
11:8	**tCYC.** Data setup and cycle time UDMA out (value + 2 cycles).
7:4	**tRP.** Ready to pause time (value + 1 cycle). Note: tRFS + 1 tRP on next clock.
3:0	**tACK.** IDE_CS[1:0]# setup to IDE_DACK0#/DACK1# (value + 1 cycle).

Core Logic Module (Continued)

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration (Continued)

Bit	Description
Index 48h-4Bh	**Channel 0 Drive 1 PIO Register (R/W)** — **Reset Value: 00009172h** **Channel 0 Drive 1 Programmed I/O Control Register.** See F2 Index 40h for bit descriptions.
Index 4Ch-4Fh	**Channel 0 Drive 1 DMA Control Register (R/W)** — **Reset Value: 00077771h** **Channel 0 Drive 1 MDMA/UDMA Control Register.** See F2 Index 44h for bit descriptions. **Note:** The PIO Mode format is selected in F2 Index 44h[31], bit 31 of this register is defined as reserved.
Index 50h-53h	**Channel 1 Drive 0 PIO Register (R/W)** — **Reset Value: 00009172h** **Channel 1 Drive 0 Programmed I/O Control Register**. See F2 Index 40h for bit descriptions.
Index 54h-57h	**Channel 1 Drive 0 DMA Control Register (R/W)** — **Reset Value: 00077771h** **Channel 1 Drive 0 MDMA/UDMA Control Register.** See F2 Index 44h for bit descriptions. **Note:** The PIO Mode format is selected in F2 Index 44h[31], bit 31 of this register is defined as reserved.
Index 58h-5Bh	**Channel 1 Drive 1 PIO Register (R/W)** — **Reset Value: 00009172h** **Channel 1 Drive 1 Programmed I/O Control Register.** See F2 Index 40h for bit descriptions.
Index 5Ch-5Fh	**Channel 1 Drive 1 DMA Control Register (R/W)** — **Reset Value: 00077771h** **Channel 1 Drive 1 MDMA/UDMA Control Register.** See F2 Index 44h for bit descriptions. **Note:** The PIO Mode format is selected in F2 Index 44h[31], bit 31 of this register is defined as reserved.
Index 60h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

5.4.3.1 IDE Controller Support Registers

F2 Index 20h, Base Address Register 4 (F2BAR4), points to the base address of where the registers for IDE controller configuration are located. Table 5-36 gives the bit formats of the I/O mapped IDE Controller Configuration registers that are accessed through F2BAR4.

Table 5-36. F2BAR4+I/O Offset: IDE Controller Configuration Registers

Bit	Description
Offset 00h	**IDE Bus Master 0 Command Register — Primary (R/W)** **Reset Value: 00h**
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the direction of bus master transfers. 0: PCI reads performed. 1: PCI writes performed. This bit should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the bus master. 0: Disable master. 1: Enable master. Bus master operations can be halted by setting this bit to 0. Once an operation has been halted, it cannot be resumed. If this bit is set to 0 while a bus master operation is active, the command is aborted and the data transferred from the drive is discarded. This bit should be reset after completion of data transfer.
Offset 01h	**Not Used**
Offset 02h	**IDE Bus Master 0 Status Register — Primary (R/W)** **Reset Value: 00h**
7	**Simplex Mode. (Read Only)** Indicates if both the primary and secondary channel operate independently. 0: Yes. 1: No (simplex mode).
6	**Drive 1 DMA Enable.** When asserted, allows Drive 1 to perform DMA transfers. 0: Disable. 1: Enable.
5	**Drive 0 DMA Enable.** When asserted, allows Drive 0 to perform DMA transfers. 0: Disable. 1: Enable.
4:3	**Reserved.** Must be set to 0. Must return 0 on reads.
2	**Bus Master Interrupt.** Indicates if the bus master detected an interrupt. 0: No. 1: Yes. Write 1 to clear.
1	**Bus Master Error.** Indicates if the bus master detected an error during data transfer. 0: No. 1: Yes. Write 1 to clear.
0	**Bus Master Active.** Indicates if the bus master is active. 0: No. 1: Yes.
Offset 03h	**Not Used**
Offset 04h-07h	**IDE Bus Master 0 PRD Table Address — Primary (R/W)** **Reset Value: 00000000h**
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for IDE Bus Master 0. When written, this field points to the first entry in a PRD table. Once IDE Bus Master 0 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this field (by adding 08h) so that is points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-36. F2BAR4+I/O Offset: IDE Controller Configuration Registers (Continued)

Bit	Description
Offset 08h	**IDE Bus Master 1 Command Register — Secondary (R/W)** **Reset Value: 00h**
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the direction of bus master transfers. 0: PCI reads are performed. 1: PCI writes are performed. This bit should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the bus master. 0: Disable master. 1: Enable master. Bus master operations can be halted by setting this bit to 0. Once an operation has been halted, it cannot be resumed. If this bit is set to 0 while a bus master operation is active, the command is aborted and the data transferred from the drive is discarded. This bit should be reset after completion of data transfer.
Offset 09h	**Not Used**
Offset 0Ah	**IDE Bus Master 1 Status Register — Secondary (R/W)** **Reset Value: 00h**
7	**Reserved. (Read Only)**
6	**Drive 1 DMA Capable.** Allow Drive 1 to perform DMA transfers. 0: Disable. 1: Enable.
5	**Drive 0 DMA Capable.** Allow Drive 0 to perform DMA transfers. 0: Disable. 1: Enable.
4:3	**Reserved.** Must be set to 0. Must return 0 on reads.
2	**Bus Master Interrupt.** Indicates if the bus master detected an interrupt. 0: No. 1: Yes. Write 1 to clear.
1	**Bus Master Error.** Indicates if the bus master detected an error during data transfer. 0: No. 1: Yes. Write 1 to clear.
0	**Bus Master Active.** Indicates if the bus master is active. 0: No. 1: Yes.
Offset 0Bh	**Not Used**
Offset 0Ch-0Fh	**IDE Bus Master 1 PRD Table Address — Secondary (R/W)** **Reset Value: 00000000h**
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for IDE Bus Master 1. When written, this field points to the first entry in a PRD table. Once IDE Bus Master 1 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this field (by adding 08h) so that is points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

5.4.4 Audio Registers - Function 3

The register designated as Function 3 (F3) is used to configure the PCI portion of support hardware for the audio registers. The bit formats for the PCI Header registers are given in Table 5-37.

A Base Address register (F3BAR0), located in the PCI Header registers of F3, is used for pointing to the register space designated for support of audio, described later in this section.

Table 5-37. F3: PCI Header Registers for Audio Configuration

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0503h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:3	**Reserved. (Read Only)**
2	**Bus Master.** Allow the Core Logic module bus mastering capabilities. 0: Disable. 1: Enable. (Default) This bit must be set to 1.
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled to access memory offsets through F3BAR0 (See F3 Index 10h).
0	**Reserved. (Read Only)**
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 040100h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register - F3BAR0 (R/W)** — **Reset Value: 00000000h** This register sets the base address of the memory mapped audio interface control register block. This is a 128-byte block of registers used to control the audio FIFO and codec interface, as well as to support VSA SMIs. Bits [11:0] are read only (0000 0000 0000), indicating a 4 KB memory address range. Refer to Table 5-38 on page 270 for the audio configuration register bit formats and reset values.
31:12	**Audio Interface Base Address**
11:0	**Address Range. (Read Only)**
Index 14h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0503h**
Index 30h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

5.4.4.1 Audio Support Registers

F3 Index 10h, Base Address Register 0 (F3BAR0), points to the base address of where the registers for audio support are located. Table 5-38 gives the bit formats of the memory mapped audio configuration registers that are accessed through F3BAR0.

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers

Bit	Description
Offset 00h-03h	**Codec GPIO Status Register (R/W)** — **Reset Value: 00000000h**
31	**Codec GPIO Interface.** 0: Disable. 1: Enable.
30	**Codec GPIO SMI.** When asserted, allows codec GPIO interrupt to generate an SMI. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[1].
29:21	**Reserved.** Must be set to 0.
20	**Codec GPIO Status Valid. (Read Only)** Indicates if the status read is valid. 0: Yes. 1: No.
19:0	**Codec GPIO Pin Status. (Read Only)** This field indicates the GPIO pin status that is received from the codec in slot 12 on the SDATA_IN signal.
Offset 04h-07h	**Codec GPIO Control Register (R/W)** — **Reset Value: 00000000h**
31:20	**Reserved.** Must be set to 0.
19:0	**Codec GPIO Pin Data.** This field indicates the GPIO pin data that is sent to the codec in slot 12 on the SDATA_OUT signal.
Offset 08h-0Bh	**Codec Status Register (R/W)** — **Reset Value: 00000000h**
31:24	**Codec Status Address. (Read Only)** Address of the register for which status is being returned. This address comes from slot 1 bits [19:12].
23	**Codec Serial INT Enable.** When asserted, allows codec serial interrupt to cause an SMI. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[1].
22	**SYNC Pin.** Sets SYNC high or low. 0: Low. 1: High.
21	**SDATA_IN2_EN.** When enabled, allows use of SDATA_IN2 input. 0: Disable. 1: Enable.
20	**Audio Bus Master 5 AC97 Slot Select.** Selects slot for Audio Bus Master 5 to receive data. 0: Slot 6. 1: Slot 11.
19	**Audio Bus Master 4 AC97 Slot Select.** Selects slot for Audio Bus Master 4 to transmit data. 0: Slot 6. 1: Slot 11.
18	**Reserved.** Must be set to 0.
17	**Status Tag. (Read Only)** The codec status data in bits [15:0] of this register is updated in the current AC97 frame. (codec ready, slot1 and slot2 bits in tag slot are all set in current AC97 frame). 0: Not new. 1: New, updated in current frame.

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
16	**Codec Status Valid. (Read Only)** Indicates if the status in bits [15:0] of this register is valid. This bit is high during slots 3 to 11 of the AC97 frame (i.e., for approximately 14.5 µs), for every frame. 0: No. 1: Yes.
15:0	**Codec Status. (Read Only)** This is the codec status data that is received from the codec in slot 2 on SDATA_IN. Only bits [19:4] are used from slot 2. If this register is read with both bits 16 and 17 of this register set to 1, this field is updated in the current AC97 frame, and codec status data is valid. This bit field is updated only if the codec sent status data.
Offset 0Ch-0Fh	**Codec Command Register (R/W)** **Reset Value: 00000000h**
31:24	**Codec Command Address.** Address of the codec control register for which the command is being sent. This address goes in slot 1 bits [19:12] on SDATA_OUT.
23:22	**Codec Communication.** Indicates the codec that the Core Logic module is communicating with. 00: Primary codec. 01: Secondary codec. 10: Third codec. 11: Fourth codec. Only 00 and 01 are valid settings for this bit field.
21:17	**Reserved.** Must be set to 0.
16	**Codec Command Valid. (Read Only)** Indicates if the command in bits [15:0] of this register is valid. 0: No. 1: Yes. This bit is set by hardware when a codec command is written to the Codec Command register. It remains set until the command has been sent to the codec.
15:0	**Codec Command.** This is the command being sent to the codec in bits [19:4] of slot 2 on SDATA_OUT.
Offset 10h-11h	**Second Level Audio SMI Status Register (RC)** **Reset Value: 0000h**
The bits in this register contain second level SMI status reporting. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. Reading this register clears the status bits at both the second and top levels. Note that bit 0 has a third level of status reporting which also must be "read to clear". A read-only “Mirror” version of this register exists at F3BAR0+I/O Memory Offset 12h. If the value of the register must be read without clearing the SMI source (and consequently de-asserting SMI), F3BAR0+Memory Offset 12h can be read instead.	
15:8	**Reserved.** Must be set to 0.
7	**Audio Bus Master 5 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 5. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 5 is enabled (F3BAR0+Memory Offset 48h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 5 SMI Status Register (F3BAR0+Memory Offset 49h[0] = 1).
6	**Audio Bus Master 4 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 4. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 4 is enabled (F3BAR0+Memory Offset 40h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 4 SMI Status Register (F3BAR0+Memory Offset 41h[0] = 1).
5	**Audio Bus Master 3 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 3. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 3 is enabled (F3BAR0+Memory Offset 38h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 3 SMI Status Register (F3BAR0+Memory Offset 39h[0] = 1).

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
4	**Audio Bus Master 2 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 2. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 2 is enabled (F3BAR0+Memory Offset 30h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 2 SMI Status Register (F3BAR0+Memory Offset 31h[0] = 1).
3	**Audio Bus Master 1 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 1. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 1 is enabled (F3BAR0+Memory Offset 28h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 1 SMI Status Register (F3BAR0+Memory Offset 29h[0] = 1).
2	**Audio Bus Master 0 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 0. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 0 is enabled (F3BAR0+Memory Offset 20h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 0 SMI Status Register (F3BAR0+Memory Offset 21h[0] = 1).
1	**Codec Serial or GPIO Interrupt SMI Status.** Indicates if an SMI was caused by a serial or GPIO interrupt from codec. 0: No. 1: Yes. SMI generation enabling for codec serial interrupt: F3BAR0+Memory Offset 08h[23] = 1. SMI generation enabling for codec GPIO interrupt: F3BAR0+Memory Offset 00h[30] = 1.
0	**I/O Trap SMI Status.** Indicates if an SMI was caused by an I/O trap. 0: No. 1: Yes. The next level (third level) of SMI status reporting is at F3BAR0+Memory Offset 14h.

Offset 12h-13h **Second Level Audio SMI Status Mirror Register (RO)** **Reset Value: 0000h**

Note: The bits in this register contain second level SMI status reporting. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. Reading this register does not clear the status bits. See F3BAR0+Memory Offset 10h.

15:8	**Reserved.** Must be set to 0.
7	**Audio Bus Master 5 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 5. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 5 is enabled (F3BAR0+Memory Offset 48h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 49h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
6	**Audio Bus Master 4 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 4. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 4 is enabled (F3BAR0+Memory Offset 40h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 41h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
5	**Audio Bus Master 3 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 3. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 3 is enabled (F3BAR0+Memory Offset 38h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 39h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
4	**Audio Bus Master 2 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 2. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 2 is enabled (F3BAR0+Memory Offset 30h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 31h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
3	**Audio Bus Master 1 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 1. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 1 is enabled (F3BAR0+Memory Offset 28h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 29h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
2	**Audio Bus Master 0 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 0. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 0 is enabled (F3BAR0+Memory Offset 20h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 21h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
1	**Codec Serial or GPIO Interrupt SMI Status.** Indicates if an SMI was caused by a serial or GPIO interrupt from codec. 0: No. 1: Yes. SMI generation enabling for codec serial interrupt: F3BAR0+Memory Offset 08h[23] = 1. SMI generation enabling for codec GPIO interrupt: F3BAR0+Memory Offset 00h[30] = 1.
0	**I/O Trap SMI Status.** Indicates if an SMI was caused by an I/O trap. 0: No. 1: Yes. The next level (third level) of SMI status reporting is at F3BAR0+Memory Offset 14h.

Offset 14h-17h **I/O Trap SMI and Fast Write Status Register (RO/RC)** **Reset Value: 00000000h**

Note: For the four SMI status bits (bits [13:10]), if the activity was a fast write to an even address, no SMI is generated regardless of the DMA, MPU, or Sound Card status. If the activity was a fast write to an odd address, an SMI is generated but bit 13 is set to a 1.

31:24	**Fast Path Write Even Access Data. (Read Only)** This bit field contains the data from the last Fast Path Write Even access. These bits change only on a fast write to an even address.
23:16	**Fast Path Write Odd Access Data. (Read Only)** This bit field contains the data from the last Fast Path Write Odd access. These bits change on a fast write to an odd address, and also on any non-fast write.
15	**Fast Write A1. (Read Only)** This bit contains the A1 value for the last Fast Write access.
14	**Read or Write I/O Access. (Read Only)** Indicates if the last trapped I/O access was a read or a write. 0: Read. 1: Write.
13	**Sound Card or FM Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the Sound Card or FM I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) Fast Path Write must be enabled, F3BAR0+Memory Offset 18h[11] = 1, for the SMI to be reported here. If Fast Path Write is disabled, the SMI is reported in bit 10 of this register. This is the third level of SMI status reporting. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[2].

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
12	**DMA Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the DMA I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) This is the third level of SMI status reporting. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[8:7].
11	**MPU Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the MPU I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) This is the third level of SMI status reporting. Second level of SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[6:5].
10	**Sound Card or FM Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the Sound Card or FM I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) Fast Path Write must be disabled, F3BAR0+Memory Offset 18h[11] = 0, for the SMI to be reported here. If Fast Path Write is enabled, the SMI is reported in bit 13 of this register. This is the third level of SMI status reporting. Second level of SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[2].
9:0	**X-Bus Address (Read Only).** This bit field] contains the captured ten bits of X-Bus address.

Offset 18h-19h	I/O Trap SMI Enable Register (R/W	)Reset Value: 0000h
15:12	**Reserved.** Must be set to 0.	
11	**Fast Path Write Enable.** Fast Path Write (an SMI is not generated on certain writes to specified addresses). 0: Disable. 1: Enable. In Fast Path Write, the Core Logic module responds to writes to addresses: 388h, 38Ah, 38B, 2x0h, 2x2h, and 2x8h.	
10:9	**Fast Read.** These two bits hold part of the response that the Core Logic module returns for reads to several I/O locations.	
8	**High DMA I/O Trap.** If this bit is enabled and an access occurs at I/O Port C0h-DFh, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[12].	
7	**Low DMA I/O Trap.** If this bit is enabled and an access occurs at I/O Port 00h-0Fh, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[12].	
6	**High MPU I/O Trap.** If this bit is enabled and an access occurs at I/O Port 330h-331h, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[11].	

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
5	**Low MPU I/O Trap.** If this bit is enabled and an access occurs at I/O Port 300h-301h, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[11].
4	**Fast Path Read Enable/SMI Disable.** When asserted, read Fast Path (an SMI is not generated on reads from specified addresses). 0: Disable. 1: Enable. In Fast Path Read the Core Logic module responds to reads of addresses: 388h-38Bh; 2x0h, 2x1, 2x2h, 2x3, 2x8 and 2x9h. If neither sound card nor FM I/O mapping is enabled, then status read trapping is not possible.
3	**FM I/O Trap.** If this bit is enabled and an access occurs at I/O Port 388h-38Bh, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0].
2	**Sound Card I/O Trap.** If this bit is enabled and an access occurs in the address ranges selected by bits [1:0], an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[10].
1:0	**Sound Card Address Range Select.** These bits select the address range for the sound card I/O trap. 00: I/O Port 220h-22Fh 01: I/O Port 240h-24Fh 10: I/O Port 260h-26Fh 11: I/O Port 280h-28Fh
Offset 1Ah-1Bh	**Internal IRQ Enable Register (R/W)** — **Reset Value: 0000h**
15	**IRQ15 Internal.** Configures IRQ15 for internal (software) or external (hardware) use. 0: External. 1: Internal.
14	**IRQ14 Internal.** Configures IRQ14 for internal (software) or external (hardware) use. 0: External. 1: Internal.
13	**Reserved.** Must be set to 0.
12	**IRQ12 Internal.** Configures IRQ12 for internal (software) or external (hardware) use. 0: External. 1: Internal.
11	**IRQ11 Internal.** Configures IRQ11 for internal (software) or external (hardware) use. 0: External. 1: Internal.
10	**IRQ10 Internal.** Configures IRQ10 for internal (software) or external (hardware) use. 0: External. 1: Internal.
9	**IRQ9 Internal.** Configures IRQ9 for internal (software) or external (hardware) use. 0: External. 1: Internal.
8	**Reserved.** Must be set to 0.
7	**IRQ7 Internal.** Configures IRQ7 for internal (software) or external (hardware) use. 0: External. 1: Internal.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
6	**Reserved.** Must be set to 0.
5	**IRQ5 Internal.** Configures IRQ5 for internal (software) or external (hardware) use. 0: External. 1: Internal.
4	**IRQ4 Internal.** Configures IRQ4 for internal (software) or external (hardware) use. 0: External. 1: Internal.
3	**IRQ3 Internal.** Configures IRQ3 for internal (software) or external (hardware) use. 0: External. 1: Internal.
2:0	**Reserved.** Must be set to 0.

Offset 1Ch-1Fh	**Internal IRQ Control Register (R/W)**	**Reset Value: 00000000h**

Note: Bits 31:16 of this register are Write Only. Reads to these bits always return a value of 0.

31	**Mask Internal IRQ15. (Write Only)** 0: Disable. 1: Enable.
30	**Mask Internal IRQ14. (Write Only)** 0: Disable. 1: Enable.
29	**Reserved. (Write Only)** Must be set to 0.
28	**Mask Internal IRQ12. (Write Only)** 0: Disable. 1: Enable.
27	**Mask Internal IRQ11. (Write Only)** 0: Disable. 1: Enable.
26	**Mask Internal IRQ10. (Write Only)** 0: Disable. 1: Enable.
25	**Mask Internal IRQ9. (Write Only)** 0: Disable. 1: Enable.
24	**Reserved. (Write Only)** Must be set to 0.
23	**Mask Internal IRQ7. (Write Only)** 0: Disable. 1: Enable.
22	**Reserved. (Write Only)** Must be set to 0.
21	**Mask Internal IRQ5. (Write Only)** 0: Disable. 1: Enable.
20	**Mask Internal IRQ4. (Write Only)** 0: Disable. 1: Enable.
19	**Mask Internal IRQ3. (Write Only)** 0: Disable. 1: Enable.
18:16	**Reserved. (Write Only)** Must be set to 0.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
15	**Assert Masked Internal IRQ15.** 0: Disable. 1: Enable.
14	**Assert Masked Internal IRQ14.** 0: Disable. 1: Enable.
13	**Reserved.** Set to 0.
12	**Assert Masked Internal IRQ12.** 0: Disable. 1: Enable.
11	**Assert masked internal IRQ11.** 0: Disable. 1: Enable.
10	**Assert Masked Internal IRQ10.** 0: Disable. 1: Enable.
9	**Assert Masked Internal IRQ9.** 0: Disable. 1: Enable.
8	**Reserved.** Set to 0.
7	**Assert Masked Internal IRQ7.** 0: Disable. 1: Enable.
6	**Reserved.** Set to 0.
5	**Assert Masked Internal IRQ5.** 0: Disable. 1: Enable.
4	**Assert Masked Internal IRQ4.** 0: Disable. 1: Enable.
3	**Assert Masked Internal IRQ3.** 0: Disable. 1: Enable.
2:0	**Reserved.** Must be set to 0.
Offset 20h	**Audio Bus Master 0 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 0: Output to codec; 32-bit; Left and Right Channels; Slots 3 and 4.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the transfer direction of the Audio Bus Master. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 0 (read), and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must either be paused, or reach EOT. Writing 0 to this bit while the bus master is operating may result in unpredictable behavior (and may crash the bus master state machine). The only recovery from such unpredictable behavior is a PCI reset.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
Offset 21h	**Audio Bus Master 0 SMI Status Register (RC)** — **Reset Value: 00h**
Audio Bus Master 0: Output to codec; 32-bit; Left and Right Channels; Slots 3 and 4.	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software has cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software has cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the bus master transferred data which is marked by EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 22h-23h	**Not Used**
Offset 24h-27h	**Audio Bus Master 0 PRD Table Address (R/W)** — **Reset Value: 00000000h**
Audio Bus Master 0: Output to codec; 32-bit; Left and Right Channels; Slots 3 and 4.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for Audio Bus Master 0. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 0 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 28h	**Audio Bus Master 1 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 1: Input from codec; 32-Bit; Left and Right Channels; Slots 3 and 4.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Set the transfer direction of Audio Bus Master 1. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 1 (write) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 1. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing this bit to 0, the bus master must be either paused or reached EOT. Writing this bit to 0 while the bus master is operating results in unpredictable behavior (and may cause a crash of the bus master state machine). The only recovery from this condition is a PCI reset.

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
Offset 29h	**Audio Bus Master 1 SMI Status Register (RC)** **Reset Value: 00h**
Audio Bus Master 1: Input from codec; 32-Bit; Left and Right Channels; Slots 3 and 4.	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software has cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software has cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the bus master transferred data which is marked by EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 2Ah-2Bh	**Not Used**
Offset 2Ch-2Fh	**Audio Bus Master 1 PRD Table Address (R/W)** **Reset Value: 00000000h**
Audio Bus Master 1: Input from codec; 32-Bit; Left and Right Channels; Slots 3 and 4.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field is a PRD table pointer for Audio Bus Master 1. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 1 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 30h	**Audio Bus Master 2 Command Register (R/W)** **Reset Value: 00h**
Audio Bus Master 2: Output to codec; 16-Bit; Slot 5.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the transfer direction of Audio Bus Master 2. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 0 (read) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 2. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or reached EOT. Writing 0 to this bit while the bus master is operating results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
Offset 31h	**Audio Bus Master 2 SMI Status Register (RC)** **Reset Value: 00h**
Audio Bus Master 2: Output to codec; 16-Bit; Slot 5.	
7:2	**Reserved**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software has cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software has cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the Bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 32h-33h	**Not Used** **Reset Value: 00h**
Offset 34h-37h	**Audio Bus Master 2 PRD Table Address (R/W)** **Reset Value: 00000000h**
Audio Bus Master 2: Output to codec; 16-Bit; Slot 5.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for Audio Bus Master 2. When written, this field points to the first entry in a PRD table. Once Audio Bus Master 2 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 38h	**Audio Bus Master 3 Command Register (R/W)** **Reset Value: 00h**
Audio Bus Master 3: Input from codec; 16-Bit; Slot 5.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the transfer direction of Audio Bus Master 3. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 1 (write) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 3. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or have reached EOT. Writing 0 to this bit while the bus master is operating results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
Offset 39h	**Audio Bus Master 3 SMI Status Register (RC)** — **Reset Value: 00h**
Audio Bus Master 3: Input from codec; 16-Bit; Slot 5.	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 3Ah-3Bh	**Not Used**
Offset 3Ch-3Fh	**Audio Bus Master 3 PRD Table Address (R/W)** — **Reset Value: 00000000h**
Audio Bus Master 3: Input from codec; 16-Bit; Slot 5.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains is a PRD table pointer for Audio Bus Master 3. When written, this field points to the first entry in a PRD table. Once Audio Bus Master 3 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 40h	**Audio Bus Master 4 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 4: Output to codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[19] selects slot).	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Set the transfer direction of Audio Bus Master 4. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 0 (read) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 4. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or have reached EOT. Writing 0 to this bit while the bus master is operating, results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
Offset 41h	**Audio Bus Master 4 SMI Status Register (RC)** — **Reset Value: 00h**
Audio Bus Master 4: Output to codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[19] selects slot).	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 42h-43h	**Not Used**
Offset 44h-47h	**Audio Bus Master 4 PRD Table Address (R/W)** — **Reset Value: 00000000h**
Audio Bus Master 4: Output to codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[19] selects slot).	
31:2	**Pointer to the Physical Region Descriptor Table.** This register is a PRD table pointer for Audio Bus Master 4. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 4 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 48h	**Audio Bus Master 5 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 5: Input from codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[20] selects slot).	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Set the transfer direction of Audio Bus Master 5. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 1 (write) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 5. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or have reached EOT. Writing 0 to this bit while the bus master is operating, results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Table 5-38. F3BAR0+Memory Offset: Audio Configuration Registers (Continued)

Bit	Description
Offset 49h	**Audio Bus Master 5 SMI Status Register (RC)** **Reset Value: 00h**
Audio Bus Master 5: Input from codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[20] selects slot).	
7:2	**Reserved**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the Bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 4Ah-4Bh	**Not Used**
Offset 4Ch-4Fh	**Audio Bus Master 5 PRD Table Address (R/W)** **Reset Value: 00000000h**
Audio Bus Master 5: Input from codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[20] selects slot).	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for Audio Bus Master 5. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 5 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.

Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs.

DWORD 0:	[31:0]	= Memory Region Physical Base Address
DWORD 1:	31	= End of Table Flag
	30	= End of Page Flag
	29	= Loop Flag (JMP)
	[28:16]	= Reserved (0)
	[15:0]	= Byte Count of the Region (Size)

Core Logic Module (Continued)

5.4.5 X-Bus Expansion Interface - Function 5

The register space designated as Function 5 (F5) is used to configure the PCI portion of support hardware for accessing the X-Bus Expansion support registers. The bit formats for the PCI Header Registers are given in Table 5-39.

Located in the PCI Header Registers of F5 are six Base Address Registers (F5BARx) used for pointing to the register spaces designated for X-Bus Expansion support, described later in this section.

Table 5-39. F5: PCI Header Registers for X-Bus Expansion

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0505h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:2	**Reserved. (Read Only)**
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. If F5BAR0, F5BAR1, F5BAR2, F5BAR3, F5BAR4, and F5BAR5 (F5 Index 10h, 14h, 18h, 1Ch, 20h, and 24h) are defined as allowing access to memory mapped registers, this bit must be set to 1. BAR configuration is programmed through the corresponding mask register (see F5 Index 40h, 44h, 48h, 4Ch, 50h, and 54h).
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycle from the PCI bus. 0: Disable. 1: Enable. If F5BAR0, F5BAR1, F5BAR2, F5BAR3, F5BAR4, and F5BAR5 (F5 Index 10h, 14h, 18h, 1Ch, 20h, and 24h) are defined as allowing access to I/O mapped registers, this bit must be set to 1. BAR configuration is programmed through the corresponding mask register (see F5 Index 40h, 44h, 48h, 4Ch, 50h, and 54h).
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 068000h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F5BAR0 (R/W)** — **Reset Value: 00000000h**
X-Bus Expansion Address Space. This register allows PCI access to I/O mapped X-Bus Expansion support registers. Bits [5:0] must be set to 000001, indicating a 64-byte aligned I/O address space. Refer to Table 5-40 on page 287 for the X-Bus Expansion configuration register bit formats and reset values. **Note:** The size and type of accessed offsets can be reprogrammed through F5BAR0 Mask Register (F5 Index 40h).	
31:6	**X-Bus Expansion Base Address.**
5:0	**Address Range.** This bit field must be set to 000001 for this register to operate correctly.
Index 14h-17h	**Base Address Register 1 - F5BAR1 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR1 Mask Register (F5 Index 44h).	
Index 18h-1Bh	**Base Address Register 2 - F5BAR2 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR1 Mask Register (F5 Index 48h).	
Index 1Ch-1Fh	**Base Address Register 3 - F5BAR3 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR3 Mask Register (F5 Index 4Ch).	

Core Logic Module (Continued)

Table 5-39. F5: PCI Header Registers for X-Bus Expansion (Continued)

Bit	Description
Index 20h-23h	**Base Address Register 4 - F5BAR4 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR4 Mask Register (F5 Index 50h).	
Index 24h-27h	**Base Address Register 5 - F5BAR5 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR5 Mask Register (F5 Index 54h).	
Index 28h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0505h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**
Index 40h-43h	**F5BAR0 Mask Address Register (R/W)** — **Reset Value: FFFFFFC1h**
To use F5BAR0, the mask register should be programmed first. The mask register defines the size of F5BAR0 and whether the accessed offset registers are memory or I/O mapped. **Note:** Whenever a value is written to this mask register, F5BAR0 must also be written (even if the value for F5BAR0 has not changed).	
Memory Base Address Register (Bit 0 = 0)	
31:4	**Address Mask.** Determines the size of the BAR. — Every bit that is a 1 is programmable in the BAR. — Every bit that is a 0 is fixed 0 in the BAR. Since the address mask goes down to bit 4, the smallest memory region is 16 bytes, however, the PCI specification suggests not using less than a 4 KB address range.
3	**Prefetchable.** Indicates whether or not the data in memory is prefetchable. This bit should be set to 1 only if all the following are true: — There are no side-effects from reads (i.e., the data at the location is not changed as a result of the read). — The device returns all bytes regardless of the byte enables. — Host bridges can merge processor writes into this range without causing errors. — The memory is not cached from the host processor. 0: Data is not prefetchable. This value is recommended if one or more of the above listed conditions is not true. 1: Data is prefetchable.
2:1	**Type.** 00: Located anywhere in the 32-bit address space. 01: Located below 1 MB. 10: Located anywhere in the 64-bit address space. 11: Reserved.
0	**This bit must be set to 0, to indicate memory base address register.**
I/O Base Address Register (Bit 0 = 1)	
31:2	**Address Mask.** Determines the size of the BAR. — Every bit that is a 1 is programmable in the BAR. — Every bit that is a 0 is fixed 0 in the BAR. Since the address mask goes down to bit 2, the smallest I/O region is 4 bytes, however, the PCI Specification suggests not using less than a 4 KB address range.
1	**Reserved.** Must be set to 0.
0	**This bit must be set to 1, to indicate an I/O base address register.**
Index 44h-47h	**F5BAR1 Mask Address Register (R/W)** — **Reset Value: 00000000h**
To use F5BAR1, the mask register should be programmed first. The mask register defines the size of F5BAR1 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions. **Note:** Whenever a value is written to this mask register, F5BAR1 must also be written (even if the value for F5BAR1 has not changed).	

Core Logic Module (Continued)

Table 5-39. F5: PCI Header Registers for X-Bus Expansion (Continued)

Bit	Description
Index 48h-4Bh	**F5BAR2 Mask Address Register (R/W)** — **Reset Value: 00000000h**
	To use F5BAR2, the mask register should be programmed first. The mask register defines the size of F5BAR2 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions.
	Note: Whenever a value is written to this mask register, F5BAR2 must also be written (even if the value for F5BAR2 has not changed).
Index 4Ch-4Fh	**F5BAR3 Mask Address Register (R/W)** — **Reset Value: 00000000h**
	To use F5BAR3, the mask register should be programmed first. The mask register defines the size of F5BAR3 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions.
	Note: Whenever a value is written to this mask register, F5BAR3 must also be written (even if the value for F5BAR3 has not changed).
Index 50h-53h	**F5BAR4 Mask Address Register (R/W)** — **Reset Value: 00000000h**
	To use F5BAR4, the mask register should be programmed first. The mask register defines the size of F5BAR4 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions.
	Note: Whenever a value is written to this mask register, F5BAR4 must also be written (even if the value for F5BAR4 has not changed).
Index 54h-57h	**F5BAR5 Mask Address Register (R/W)** — **Reset Value: 00000000h**
	To use F5BAR5, the mask register should be programmed first. The mask register defines the size of F5BAR5 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions.
	Note: Whenever a value is written to this mask register, F5BAR5 must also be written (even if the value for F5BAR5 has not changed).
Index 58h	**F5BARx Initialized Register (R/W)** — **Reset Value: 00h**
7:6	**Reserved.** Must be set to 0.
5	**F5BAR5 Initialized.** This bit indicates if F5BAR5 (F5 Index 24h) has been initialized. At reset this bit is cleared (0). Writing F5BAR5 sets this bit to 1. If this bit programmed to 0, the decoding of F5BAR5 is disabled until either this bit is set to 1 or F5BAR5 is written (which causes this bit to be set to 1).
4	**F5BAR4 Initialized.** This bit indicates if F5BAR4 (F5 Index 28h) has been initialized. At reset this bit is cleared (0). Writing F5BAR4 sets this bit to 1. If this bit programmed to 0, the decoding of F5BAR4 is disabled until either this bit is set to 1 or F5BAR4 is written (which causes this bit to be set to 1).
3	**F5BAR3 Initialized.** This bit indicates if F5BAR3 (F5 Index 1Ch) has been initialized. At reset this bit is cleared (0). Writing F5BAR3 sets this bit to 1. If this bit programmed to 0, the decoding of F5BAR3 is disabled until either this bit is set to 1 or F5BAR3 is written (which causes this bit to be set to 1).
2	**F5BAR2 Initialized.** This bit indicates if F5BAR2 (F5 Index 18h) has been initialized. At reset this bit is cleared (0). Writing F5BAR2 sets this bit to 1. If this bit programmed to 0, the decoding of F5BAR2 is disabled until either this bit is set to 1 or F5BAR2 is written (which causes this bit to be set to 1).
1	**F5BAR1 Initialized.** This bit indicates if F5BAR1 (F5 Index 14h) has been initialized. At reset this bit is cleared (0). Writing F5BAR1 sets this bit to 1. If this bit programmed to 0, the decoding of F5BAR1 is disabled until either this bit is set to 1 or F5BAR1 is written (which causes this bit to be set to 1).
0	**F5BAR0 Initialized.** This bit indicates if F5BAR0 (F5 Index 10h) has been initialized. At reset this bit is cleared (0). Writing F5BAR0 sets this bit to 1. If this bit programmed to 0, the decoding of F5BAR0 is disabled until either this bit is set to 1 or F5BAR0 is written (which causes this bit to be set to 1).
Index 59h-5Fh	**Reserved** — **Reset Value: xxh**
Index 60h-63h	**Scratchpad: Usually used for Device Number (R/W)** — **Reset Value: 00000000h**
	BIOS writes a value, of the Device number. Expected value: 00002200h.
Index 64h-67h	**Scratchpad: Usually used for Configuration Block Address (R/W)** — **Reset Value: 00000000h**
	BIOS writes a value, of the Configuration Block Address.
Index 68h-FFh	**Reserved**

Core Logic Module (Continued)

5.4.5.1 X-Bus Expansion Support Registers

F5 Index 10h, Base Address Register 0 (F5BAR0) set the base address that allows PCI access to additional I/O Control support registers. Table 5-40 shows the support registers accessed through F5BAR0.

Table 5-40. F5BAR0+I/O Offset: X-Bus Expansion Registers

Bit	Description
Offset 00h-03h	**I/O Control Register 1 (R/W)** — **Reset Value: 010C0007h**
31:28	**Reserved.**
27	**IO_ENABLE_SIO_IR (Enable Integrated SIO Infrared).** 0: Disable. 1: Enable.
26:25	**IO_SIOCFG_IN (Integrated SIO Input Configuration).** These two bits can be used to disable the integrated SIO totally or limit/control the base address. 00: Integrated SIO disable. 01: Integrated SIO configuration access disable. 10: Integrated SIO base address 02Eh/02Fh enable. 11: Integrated SIO base address 015Ch/015Dh enable.
24	**IO_ENABLE_SIO_DRIVING_ISA_BUS (Enable Integrated SIO ISA Bus Control).** Allow the integrated SIO to drive the internal ISA bus. 0: Disable. 1: Enable. (Default)
23:21	**Reserved.** Set to 0.
20	**IO_USB_SMI_PWM_EN (USB Internal SMI).** Route USB-generated SMI to SMI Status Register in F1BAR0+I/O Offset 00h/02h[14]. 0: Disable. 1: Enable.
19	**IO_USB_SMI_EN (USB SMI Configuration).** Allow USB-generated SMIs. 0: Disable. 1: Enable. If bits 19 and 20 are enabled, the SMI generated by the USB is reported via the Top Level SMI status register at F1BAR0+I/O Offset 00h/02h[14]. If only bit 19 is enabled, the USB can generate an SMI but there is no status reporting.
18	**IO_USB_PCI_EN (USB).** Enables USB ports. 0: Disable. 1: Enable.
17:0	**Reserved.**
Offset 04h-07h	**I/O Control Register 2 (R/W)** — **Reset Value: 00000002h**
31:2	**Reserved.** Write as read.
1	**Video Processor Access Enable.** Allows access to video processor using F4BAR0. 0: Disable. 1: Enable. (Default) **Note:** This bit is readable after the register (F5BAR0+Offset 04h) has been written once.
0	**IO_STRAP_IDSEL_SELECT (IDSEL Strap Override).** 0: IDSEL: AD28 for Chipset Register Space (F0-F5), AD29 for USB Register Space (PCIUSB). 1: IDSEL: AD26 for Chipset Register Space (F0-F5), AD27 for USB Register Space (PCIUSB).

Core Logic Module (Continued)

Table 5-40. F5BAR0+I/O Offset: X-Bus Expansion Registers (Continued)

Bit	Description
Offset 08h-0Bh	**I/O Control Register 3 (R/W)** **Reset Value: 00009000h**
31:16	**Reserved.** Write as read.
15:13	**IO_USB_XCVR_VADJ (USB Voltage Adjustment Connection).** These bits connect to the voltage adjustment interface on the three USB transceivers. Default = 100.
12:8	**IO_USB_XCVT_CADJ (USB Current Adjustment).** These bits connect to the current adjustment interface on the three USB transceivers. Default = 10000.
7	**IO_TEST_PORT_EN (Debug Test Port Enable).** 0: Disable. 1: Enable.
6:0	**IO_TEST_PORT_REG (Debug Port Pointer).** These bits are used to point to the 16-bit slice of the test port bus.

Core Logic Module (Continued)

5.4.6 USB Controller Registers - PCIUSB

The registers designated as PCIUSB are 32-bit registers decoded from the PCI address bits [7:2] and C/BE[3:0]#, when IDSEL is high, AD[10:8] select the appropriate function, and AD[1:0] are 00.

The PCI Configuration registers are listed in Table 5-41. They can be accessed as any number of bytes within a single 32-bit aligned unit. They are selected by the PCI-standard Index and Byte-Enable method.

In the PCI Configuration space, there is one Base Address Register (BAR), at Index 10h, which is used to map the USB Host Controller's operational register set into a 4K memory space. Once the BAR register has been initialized, and the PCI Command register at Index 04h has been set to enable the Memory space decoder, these “USB Controller” registers are accessible.

The memory-mapped USB Controller registers are listed in Table 5-42. They follow the Open Host Controller Interface (OHCI) specification. Registers marked as “Reserved”, and reserved bits within a register, should not be changed by software.

Table 5-41. PCIUSB: USB PCI Configuration Registers

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 0E11h**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: A0F8h**
Index 04h-05h	**Command Register (R/W)** — **Reset Value: 00h**
15:10	**Reserved.** Must be set to 0.
9	**Fast Back-to-Back Enable. (Read Only)** USB only acts as a master to a single device, so this functionality is not needed. It is always disabled (i.e., this bit must always be set to 0).
8	**SERR#.** When this bit is enabled, USB asserts SERR# when it detects an address parity error. 0: Disable. 1: Enable.
7	**Wait Cycle Control.** USB does not need to insert a wait state between the address and data on the AD lines. It is always disabled (i.e., this bit is set to 0).
6	**Parity Error.** USB asserts PERR# when it is the agent receiving data and it detects a data parity error. 0: Disable. 1: Enable.
5	**VGA Palette Snoop Enable. (Read Only)** USB does not support this function. It is always disabled (i.e., this bit is set to 0).
4	**Memory Write and Invalidate.** Allow USB to run Memory Write and Invalidate commands. 0: Disable. 1: Enable. The Memory Write and Invalidate Command only occurs if the cache-line size is set to 32 bytes and the memory write is exactly one cache line. This bit must be set to 0.
3	**Special Cycles.** USB does not run special cycles on PCI. It is always disabled (i.e., this bit is set to 0).
2	**PCI Master Enable**. Allow the USB to run PCI master cycles. 0: Disable. 1: Enable.
1	**Memory Space.** Allow the USB to respond as a target to memory cycles from the PCI bus. 0: Disable. 1: Enable.
0	**I/O Space.** Allow the USB to respond as a target to I/O cycles from the PCI bus. 0 Disable. 1: Enable.

Core Logic Module (Continued)

Table 5-41. PCIUSB: USB PCI Configuration Registers (Continued)

Bit	Description
Index 06h-07h	**Status Register (R/W)** — **Reset Value: 0280h**
The PCI specification defines this register to record status information for PCI related events. This is a read/write register. However, writes can only reset bits. A bit is reset whenever the register is written and the data in the corresponding bit location is a 1.	
15	**Detected Parity Error**. This bit is set to 1 whenever the USB detects a parity error, even if the Parity Error (Response) Detection Enable Bit (Command Register, bit 6) is disabled. Write 1 to clear.
14	**SERR# Status.** This bit is set whenever the USB detects a PCI address error. Write 1 to clear.
13	**Received Master Abort Status.** This bit is set when the USB, acting as a PCI master, aborts a PCI bus memory cycle. Write 1 to clear.
12	**Received Target Abort Status.** This bit is set when a USB generated PCI cycle (USB is the PCI master) is aborted by a PCI target. Write 1 to clear.
11	**Signaled Target Abort Status.** This bit is set whenever the USB signals a target abort. Write 1 to clear.
10:9	**DEVSEL# Timing. (Read Only)** These bits indicate the DEVSEL# timing when performing a positive decode. Since DEVSEL# is asserted to meet the medium timing, these bits are encoded as 01b.
8	**Data Parity Reported. (Read Only)** This bit is set to 1 if the Parity Error Response bit (Command Register bit 6) is set, and the USB detects PERR# asserted while acting as PCI master (whether or not PERR# was driven by USB).
7	**Fast Back-to-Back Capable.** The USB supports fast back-to-back transactions when the transactions are not to the same agent. This bit is always 1.
6:0	**Reserved**. Must be set to 0.
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 08h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 0C0310h**
This register identifies the generic function of the USB the specific register level programming interface. The Base Class is 0Ch (Serial Bus Controller). The Sub Class is 03h (Universal Serial Bus). The Programming Interface is 10h (OpenHCI).	
Index 0Ch	**Cache Line Size Register (R/W)** — **Reset Value: 00h**
This register identifies the system cache-line size in units of 32-bit WORDs. The USB only stores the value of bit 3 in this register since the cache-line size of 32 bytes is the only value applicable to the design. Any value other than 08h written to this register is read back as 00h.	
Index 0Dh	**Latency Timer Register (R/W)** — **Reset Value: 00h**
This register identifies the value of the latency timer in PCI clocks for PCI bus master cycles. Bits [2:0] of this register are always set to 0.	
Index 0Eh	**Header Type Register (RO)** — **Reset Value: 00h**
This register identifies the type of the predefined header in the configuration space. Since the USB is a single function device and not a PCI-to-PCI bridge, this byte should be read as 00h.	
Index 0Fh	**BIST Register (RO)** — **Reset Value: 00h**
This register identifies the control and status of Built-In Self-Test (BIST). The USB does not implement BIST, so this register is read only.	
Index 10h-13h	**Base Address Register- USB_BAR0 (R/W)** — **Reset Value: 00000000h**
31:12	**Base Address.** POST writes the value of the memory base address to this register.
11:4	**Always 0.** Indicates that a 4 KB address range is requested.
3	**Always 0.** Indicates that there is no support for prefetchable memory.
2:1	**Always 0.** Indicates that the base register is 32-bits wide and can be placed anywhere in 32-bit memory space.
0	**Always 0.** Indicates that the operational registers are mapped into memory space.

Core Logic Module (Continued)

Table 5-41. PCIUSB: USB PCI Configuration Registers (Continued)

Bit	Description
Index 14h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 0E11h**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: A0F8h**
Index 30h-3Bh	**Reserved** — **Reset Value: 00h**
Index 3Ch	**Interrupt Line Register (R/W)** — **Reset Value: 00h**
This register identifies the system interrupt controllers to which the device's interrupt pin is connected. The value of this register is used by device drivers and has no direct meaning to USB.	
Index 3Dh	**Interrupt Pin Register (R/W)** — **Reset Value: 01h**
This register selects which interrupt pin the device uses. USB uses INTA# after reset. INTB#, INTC# or INTD# can be selected by writing 2, 3 or 4, respectively.	
Index 3Eh	**Min. Grant Register (RO)** — **Reset Value: 00h**
This register specifies how long a burst is needed by the USB, assuming a clock rate of 33 MHz. The value in this register specifies a period of time in units of 1/4 microsecond.	
Index 3Fh	**Max. Latency Register (RO)** — **Reset Value: 50h**
This register specifies how often (in units of 1/4 microsecond) the USB needs access to the PCI bus assuming a clock rate of 33 MHz.	
Index 40h-43h	**ASIC Test Mode Enable Register (R/W)** — **Reset Value: 000F0000h**
Used for internal debug and test purposes only.	
Index 44h	**ASIC Operational Mode Enable Register (R/W)** — **Reset Value: 00h**
7:1	**Write Only.** Read as 0s.
0	**Data Buffer Region 16** 0: The size of the region for the data buffer is 32 bytes. 1: The size of the region for the data buffer is 16 bytes.
Index 45h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers

Bit	Description
Offset 00h-03h	**HcRevision Register (RO)** — **Reset Value = 00000110h**
31:8	**Reserved.** Read/Write 0s.
7:0	**Revision (Read Only).** Indicates the Open HCI Specification revision number implemented by the Hardware. USB supports 1.0 specification. (X.Y = XYh).
Offset 04h-07h	**HcControl Register (R/W)** — **Reset Value = 00000000h**
31:11	**Reserved.** Read/Write 0s.
10	**RemoteWakeupConnectedEnable.** If a remote wakeup signal is supported, this bit enables that operation. Since there is no remote wakeup signal supported, this bit is ignored.
9	**RemoteWakeupConnected (Read Only).** This bit indicated whether the HC supports a remote wakeup signal. This implementation does not support any such signal. The bit is hard-coded to 0.
8	**InterruptRouting.** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
7:6	**HostControllerFunctionalState.** This field sets the HC state. The HC may force a state change from UsbSuspend to UsbResume after detecting resume signaling from a downstream port. States are: 00: UsbReset. 01: UsbResume. 10: UsbOperational. 11: UsbSuspend.
5	**BulkListEnable.** When set, this bit enables processing of the Bulk list.
4	**ControlListEnable.** When set, this bit enables processing of the Control list.
3	**IsochronousEnable.** When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the HC will check this bit when it finds an isochronous ED.
2	**PeriodicListEnable.** When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The HC checks this bit prior to attempting any periodic transfers in a frame.
1:0	**ControlBulkServiceRatio.** Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e., 00: 1 Control Endpoint; 11: 3 Control Endpoints).
Offset 08h-0Bh	**HcCommandStatus Register (R/W)** — **Reset Value = 00000000h**
31:18	**Reserved.** Read/Write 0s.
17:16	**ScheduleOverrunCount.** This field increments every time the SchedulingOverrun bit in HcInterruptStatus is set. The count wraps from 11 to 00.
15:4	**Reserved.** Read/Write 0s.
3	**OwnershipChangeRequest.** When set by software, this bit sets the OwnershipChange field in HcInterruptStatus. The bit is cleared by software.
2	**BulkListFilled.** Set to indicate there is an active ED on the Bulk List. The bit may be set by either software or the HC and cleared by the HC each time it begins processing the head of the Bulk List.
1	**ControlListFilled.** Set to indicate there is an active ED on the Control List. It may be set by either software or the HC and cleared by the HC each time it begins processing the head of the Control List.
0	**HostControllerReset.** This bit is set to initiate a software reset. This bit is cleared by the HC upon completion of the reset operation.
Offset 0Ch-0Fh	**HcInterruptStatus Register (R/W)** — **Reset Value = 00000000h**
31	**Reserved.** Read/Write 0s.
30	**OwnershipChange.** This bit is set when the OwnershipChangeRequest bit of HcCommandStatus is set.
29:7	**Reserved.** Read/Write 0s.
6	**RootHubStatusChange.** This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus register has changed.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
5	**FrameNumberOverflow.** Set when bit 15 of FrameNumber changes value.
4	**UnrecoverableError (Read Only).** This event is not implemented and is hard-coded to 0. Writes are ignored.
3	**ResumeDetected.** Set when HC detects resume signaling on a downstream port.
2	**StartOfFrame.** Set when the Frame Management block signals a Start of Frame event.
1	**WritebackDoneHead.** Set after the HC has written HcDoneHead to HccaDoneHead.
0	**SchedulingOverrun.** Set when the List Processor determines a Schedule Overrun has occurred.
Note:	All bits are set by hardware and cleared by software.

Offset 10h-13h	**HcInterruptEnable Register (R/W)**	**Reset Value = 00000000h**
31	**MasterInterruptEnable.** This bit is a global interrupt enable. A write of 1 allows interrupts to be enabled via the specific enable bits listed above.	
30	**OwnershipChangeEnable.** 0: Ignore. 1: Enable interrupt generation due to Ownership Change.	
29:7	**Reserved.** Read/Write 0s.	
6	**RootHubStatusChangeEnable.** 0: Ignore. 1: Enable interrupt generation due to Root Hub Status Change.	
5	**FrameNumberOverflowEnable.** 0: Ignore. 1: Enable interrupt generation due to Frame Number Overflow.	
4	**UnrecoverableErrorEnable.** This event is not implemented. All writes to this bit are ignored.	
3	**ResumeDetectedEnable.** 0: Ignore. 1: Enable interrupt generation due to Resume Detected.	
2	**StartOfFrameEnable.** 0: Ignore. 1: Enable interrupt generation due to Start of Frame.	
1	**WritebackDoneHeadEnable.** 0: Ignore. 1: Enable interrupt generation due to Writeback Done Head.	
0	**SchedulingOverrunEnable.** 0: Ignore. 1: Enable interrupt generation due to Scheduling Overrun.	
Note:	Writing a 1 to a bit in this register sets the corresponding bit, while writing a 0 leaves the bit unchanged.	

Offset 14h-17h	**HcInterruptDisable Register (R/W)**	**Reset Value = 00000000h**
31	**MasterInterruptEnable.** Global interrupt disable. A write of 1 disables all interrupts.	
30	**OwnershipChangeEnable.** 0: Ignore. 1: Disable interrupt generation due to Ownership Change.	
29:7	**Reserved.** Read/Write 0s.	
6	**RootHubStatusChangeEnable.** 0: Ignore. 1: Disable interrupt generation due to Root Hub Status Change.	

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
5	**FrameNumberOverflowEnable.** 0: Ignore. 1: Disable interrupt generation due to Frame Number Overflow.
4	**UnrecoverableErrorEnable.** This event is not implemented. All writes to this bit will be ignored.
3	**ResumeDetectedEnable.** 0: Ignore. 1: Disable interrupt generation due to Resume Detected.
2	**StartOfFrameEnable.** 0: Ignore. 1: Disable interrupt generation due to Start of Frame.
1	**WritebackDoneHeadEnable.** 0: Ignore. 1: Disable interrupt generation due to Writeback Done Head.
0	**SchedulingOverrunEnable.** 0: Ignore. 1: Disable interrupt generation due to Scheduling Overrun.
Note: Writing a 1 to a bit in this register clears the corresponding bit, while writing a 0 to a bit leaves the bit unchanged.	
Offset 18h-1Bh	**HcHCCA Register (R/W)** — **Reset Value = 00000000h**
31:8	**HCCA.** Pointer to HCCA base address.
7:0	**Reserved.** Read/Write 0s.
Offset 1Ch-1Fh	**HcPeriodCurrentED Register (R/W)** — **Reset Value = 00000000h**
31:4	**PeriodCurrentED.** Pointer to the current Periodic List ED.
3:0	**Reserved.** Read/Write 0s.
Offset 20h-23h	**HcControlHeadED Register (R/W)** — **Reset Value = 00000000h**
31:4	**ControlHeadED.** Pointer to the Control List Head ED.
3:0	**Reserved.** Read/Write 0s.
Offset 24h-27h	**HcControlCurrentED Register (R/W)** — **Reset Value = 00000000h**
31:4	**ControlCurrentED.** Pointer to the current Control List ED.
3:0	**Reserved.** Read/Write 0s.
Offset 28h-2Bh	**HcBulkHeadED Register (R/W)** — **Reset Value = 00000000h**
31:4	**BulkHeadED.** Pointer to the Bulk List Head ED.
3:0	**Reserved.** Read/Write 0s.
Offset 2Ch-2Fh	**HcBulkCurrentED Register (R/W)** — **Reset Value = 00000000h**
31:4	**BulkCurrentED.** Pointer to the current Bulk List ED.
3:0	**Reserved.** Read/Write 0s.
Offset 30h-33h	**HcDoneHead Register (R/W)** — **Reset Value = 00000000h**
31:4	**DoneHead.** Pointer to the current Done List Head ED.
3:0	**Reserved.** Read/Write 0s.
Offset 34h-37h	**HcFmInterval Register (R/W)** — **Reset Value = 00002EDFh**
31	**FrameIntervalToggle (Read Only).** This bit is toggled by HCD when it loads a new value into FrameInterval.
30:16	**FSLargestDataPacket (Read Only).** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
15:14	**Reserved.** Read/Write 0s.
13:0	**FrameInterval.** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.
Offset 38h-3Bh	**HcFrameRemaining Register (RO)** **Reset Value = 00000000h**
31	**FrameRemainingToggle (Read Only).** Loaded with FrameIntervalToggle when FrameRemaining is loaded.
30:14	**Reserved.** Read 0s.
13:0	**FrameRemaining (Read Only).** When the HC is in the UsbOperational state, this 14-bit field decrements each 12 MHz clock period. When the count reaches 0, (end of frame) the counter reloads with FrameInterval. In addition, the counter loads when the HC transitions into UsbOperational.
Offset 3Ch-3Fh	**HcFmNumber Register (RO)** **Reset Value = 00000000h**
31:16	**Reserved.** Read 0s.
15:0	**FrameNumber (Read Only).** This 16-bit incrementing counter field is incremented coincident with the loading of FrameRemaining. The count rolls over from FFFFh to 0h.
Offset 40h-43h	**HcPeriodicStart Register (R/W)** **Reset Value = 00000000h**
31:14	**Reserved.** Read/Write 0s.
13:0	**PeriodicStart.** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.
Offset 44h-47h	**HcLSThreshold Register (R/W)** **Reset Value = 00000628h**
31:12	**Reserved.** Read/Write 0s.
11:0	**LSThreshold.** This field contains a value used by the Frame Management block to determine whether or not a low speed transaction can be started in the current frame.
Offset 48h-4Bh	**HcRhDescriptorA Register (R/W)** **Reset Value = 01000003h**
31:24	**PowerOnToPowerGoodTime.** This field value is represented as the number of 2 ms intervals, ensuring that the power switching is effective within 2 ms. Only bits [25:24] are implemented as R/W. The remaining bits are read only as 0. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should be written to support system implementation. This field should always be written to a non-zero value.
23:13	**Reserved.** Read/Write 0s.
12	**NoOverCurrentProtection.** This bit should be written to support the external system port over-current implementation. 0: Over-current status is reported. 1: Over-current status is not reported.
11	**OverCurrentProtectionMode.** This bit should be written 0 and is only valid when NoOverCurrentProtection is cleared. 0: Global Over-Current. 1: Individual Over-Current.
10	**DeviceType (Read Only).** USB is not a compound device.
9	**NoPowerSwitching.** This bit should be written to support the external system port power switching implementation. 0: Ports are power switched. 1: Ports are always powered on.
8	**PowerSwitchingMode.** This bit is only valid when NoPowerSwitching is cleared. This bit should be written 0. 0: Global Switching. 1: Individual Switching.
7:0	**NumberDownstreamPorts (Read Only).** USB supports three downstream ports.
Note:	This register is only reset by a power-on reset (PCIRST#). It is written during system initialization to configure the Root Hub. These bit should not be written during normal operation.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 4Ch-4Fh	**HcRhDescriptorB Register (R/W)** — **Reset Value = 00000000h**
31:16	**PortPowerControlMask.** Global-power switching. This field is only valid if NoPowerSwitching is cleared and PowerSwitchingMode is set (individual port switching). When set, the port only responds to individual port power switching commands (Set/ClearPortPower). When cleared, the port only responds to global power switching commands (Set/ClearGlobalPower). 0: Device not removable. 1: Global-power mask. Port Bit relationship - Unimplemented ports are reserved, read/write 0. 0 = Reserved 1 = Port 1 2 = Port 2 ... 15 = Port 15
15:0	**DeviceRemoveable.** USB ports default to removable devices. 0: Device not removable. 1: Device removable. Port Bit relationship 0 = Reserved 1 = Port 1 2 = Port 2 ... 15 = Port 15 Unimplemented ports are reserved, read/write 0.
Note:	This register is only reset by a power-on reset (PCIRST#). It is written during system initialization to configure the Root Hub. These bit should not be written during normal operation.
Offset 50h-53h	**HcRhStatus Register (R/W)** — **Reset Value = 00000000h**
31	**ClearRemoteWakeupEnable (Write Only).** Writing a 1 to this bit clears DeviceRemoteWakeupEnable. Writing a 0 has no effect.
30:18	**Reserved.** Read/Write 0s.
17	**OverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
16	**Read: LocalPowerStatusChange.** Not supported. Always read 0. **Write: SetGlobalPower.** Write a 1 issues a SetGlobalPower command to the ports. Writing a 0 has no effect.
15	**Read: DeviceRemoteWakeupEnable.** This bit enables ports' ConnectStatusChange as a remote wakeup event. 0: Disabled. 1: Enabled. **Write: SetRemoteWakeupEnable.** Writing a 1 sets DeviceRemoteWakeupEnable. Writing a 0 has no effect.
14:2	**Reserved.** Read/Write 0s.
1	**OverCurrentIndicator.** This bit reflects the state of the OVRCUR pin. This field is only valid if NoOverCurrentProtection and OverCurrentProtectionMode are cleared. 0: No over-current condition. 1: Over-current condition.
0	**Read: LocalPowerStatus.** Not Supported. Always read 0. **Write: ClearGlobalPower.** Writing a 1 issues a ClearGlobalPower command to the ports. Writing a 0 has no effect.
Note:	This register is reset by the UsbReset state.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 54h-57h	**HcRhPortStatus[1] Register (R/W)** — **Reset Value = 00000000h**
31:21	**Reserved.** Read/Write 0s.
20	**PortResetStatusChange.** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	**PortOverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
18	**PortSuspendStatusChange.** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	**PortEnableStatusChange.** This bit indicates that the port has been disabled due to a hardware event (cleared PortEnableStatus). 0: Port has not been disabled. 1: PortEnableStatus has been cleared.
16	**ConnectStatusChange.** This bit indicates a connect or disconnect event has been detected. Writing a 1 clears this bit. Writing a 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. If DeviceRemoveable is set, this bit resets to 1.
15:10	**Reserved.** Read/Write 0s.
9	**Read: LowSpeedDeviceAttached.** This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full Speed device. 1: Low Speed device. **Write: ClearPortPower.** Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.
8	**Read: PortPowerStatus.** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. If NoPowerSwitching is set, this bit is always read as 1. **Write: SetPortPower.** Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.
7:5	**Reserved.** Read/Write 0s.
4	**Read: PortResetStatus.** 0: Port reset signal is not active. 1: Port reset signal is active. **Write: SetPortReset.** Writing a 1 sets PortResetStatus. Writing a 0 has no effect.
3	**Read: PortOverCurrentIndicator.** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition. 1: Over-current condition. **Write: ClearPortSuspend.** Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.
2	**Read: PortSuspendStatus.** 0: Port is not suspended. 1: Port is selectively suspended. **Write: SetPortSuspend.** Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.
1	**Read: PortEnableStatus.** 0: Port disabled. 1: Port enabled. **Write: SetPortEnable.** Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
0	**Read: CurrentConnectStatus.** 0: No device connected. 1: Device connected. If DeviceRemoveable is set (not removable) this bit is always 1. **Write: ClearPortEnable.** Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.
Note: This register is reset by the UsbReset state.	
Offset 58h-5Bh	**HcRhPortStatus[2] Register (R/W)** **Reset Value = 00000000h**
31:21	**Reserved.** Read/Write 0s.
20	**PortResetStatusChange.** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	**PortOverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
18	**PortSuspendStatusChange.** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	**PortEnableStatusChange.** This bit indicates that the port has been disabled due to a hardware event (cleared PortEnableStatus). 0: Port has not been disabled. 1: PortEnableStatus has been cleared.
16	**ConnectStatusChange.** This bit indicates a connect or disconnect event has been detected. Writing a 1 clears this bit. Writing a 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. If DeviceRemoveable is set, this bit resets to 1.
15:10	**Reserved.** Read/Write 0s.
9	**Read: LowSpeedDeviceAttached.** This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full speed device. 1: Low speed device. **Write: ClearPortPower.** Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.
8	**Read: PortPowerStatus.** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. If NoPowerSwitching is set, this bit is always read as 1. **Write: SetPortPower.** Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.
7:5	**Reserved.** Read/Write 0s.
4	**Read: PortResetStatus.** 0: Port reset signal is not active. 1: Port reset signal is active. **Write: SetPortReset.** Writing a 1 sets PortResetStatus. Writing a 0 has no effect.
3	**Read: PortOverCurrentIndicator.** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition. 1: Over-current condition. **Write: ClearPortSuspend.** Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
2	**Read: PortSuspendStatus.** 0: Port is not suspended. 1: Port is selectively suspended. **Write: SetPortSuspend.** Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.
1	**Read: PortEnableStatus.** 0: Port disabled. 1: Port enabled. **Write: SetPortEnable.** Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.
0	**Read: CurrentConnectStatus.** 0: No device connected. 1: Device connected. If DeviceRemoveable is set (not removable) this bit is always 1. **Write: ClearPortEnable.** Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.
Note: This register is reset by the UsbReset state.	

Offset 5Ch-5Fh	HcRhPortStatus[3] Register (R/W)	Reset Value = 00000000h
31:21	**Reserved.** Read/Write 0s.	
20	**PortResetStatusChange.** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.	
19	**PortOverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.	
18	**PortSuspendStatusChange.** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.	
17	**PortEnableStatusChange.** This bit indicates that the port has been disabled due to a hardware event (cleared PortEnableStatus). 0: Port has not been disabled. 1: PortEnableStatus has been cleared.	
16	**ConnectStatusChange.** This bit indicates a connect or disconnect event has been detected. Writing a 1 clears this bit. Writing a 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. If DeviceRemoveable is set, this bit resets to 1.	
15:10	**Reserved.** Read/Write 0s.	
9	**Read: LowSpeedDeviceAttached.** This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full speed device. 1: Low speed device. **Write: ClearPortPower.** Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.	
8	**Read: PortPowerStatus.** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. If NoPowerSwitching is set, this bit is always read as 1. **Write: SetPortPower.** Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.	
7:5	**Reserved.** Read/Write 0s.	

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
4	**Read: PortResetStatus.** 0: Port reset signal is not active. 1: Port reset signal is active. **Write: SetPortReset.** Writing a 1 sets PortResetStatus. Writing a 0 has no effect.
3	**Read: PortOverCurrentIndicator.** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition. 1: Over-current condition. **Write: ClearPortSuspend.** Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.
2	**Read: PortSuspendStatus.** 0: Port is not suspended. 1: Port is selectively suspended. **Write: SetPortSuspend.** Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.
1	**Read: PortEnableStatus.** 0: Port disabled. 1: Port enabled. **Write: SetPortEnable.** Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.
0	**Read: CurrentConnectStatus.** 0: No device connected. 1: Device connected. If DeviceRemoveable is set (not removable) this bit is always 1. **Write: ClearPortEnable.** Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.
Note:	This register is reset by the UsbReset state.

Offset 60h-9Fh	**Reserved**	**Reset Value = xxh**

Offset 100h-103h	**HceControl Register (R/W)**	**Reset Value = 00000000h**

31:9	**Reserved.** Read/Write 0s.
8	**A20State.** Indicates current state of Gate A20 on keyboard controller. Compared against value written to 60h when GateA20Sequence is active.
7	**IRQ12Active.** Indicates a positive transition on IRQ12 from keyboard controller occurred. Software writes this bit to 1 to clear it (set it to 0); a 0 write has no effect.
6	**IRQ1Active.** Indicates a positive transition on IRQ1 from keyboard controller occurred. Software writes this bit to 1 to clear it (set it to 0); a 0 write has no effect.
5	**GateA20Sequence.** Set by HC when a data value of D1h is written to I/O port 64h. Cleared by HC on write to I/O port 64h of any value other than D1h.
4	**ExternalIRQEn.** When set to 1, IRQ1 and IRQ12 from the keyboard controller cause an emulation interrupt. The function controlled by this bit is independent of the setting of the EmulationEnable bit in this register.
3	**IRQEn.** When set, the HC generates IRQ1 or IRQ12 as long as the OutputFull bit in HceStatus is set to 1. If the AuxOutputFull bit of HceStatus is 0, IRQ1 is generated: if 1, then an IRQ12 is generated.
2	**CharacterPending.** When set, an emulation interrupt will be generated when the OutputFull bit of the HceStatus register is set to 0.
1	**EmulationInterrupt (Read Only).** This bit is a static decode of the emulation interrupt condition.
0	**EmulationEnable.** When set to 1 the HC is enabled for legacy emulation and will decode accesses to I/O registers 60h and 64h and generate IRQ1 and/or IRQ12 when appropriate. The HC also generates an emulation interrupt at appropriate times to invoke the emulation software.
Note:	This register is used to enable and control the emulation hardware and report various status information.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 104h-107h	**HceInput Register (R/W)** — **Reset Value = 000000xxh**
31:8	**Reserved.** Read/Write 0s.
7:0	**InputData.** This register holds data written to I/O ports 60h and 64h.
Note:	This register is the emulation side of the legacy Input Buffer register.
Offset 108h-10Bh	**HceOutput Register (R/W)** — **Reset Value = 000000xxh**
31:8	**Reserved.** Read/Write 0s.
7:0	**OutputData.** This register hosts data that is returned when an I/O read of port 60h is performed by application software.
Note:	This register is the emulation side of the legacy Output Buffer register where keyboard and mouse data is to be written by software.
Offset 10Ch-10Fh	**HceStatus Register (R/W)** — **Reset Value = 00000000h**
31:8	**Reserved.** Read/Write 0s.
7	**Parity.** Indicates parity error on keyboard/mouse data.
6	**Timeout.** Used to indicate a timeout.
5	**AuxOutputFull.** IRQ12 is asserted whenever this bit is set to 1 and OutputFull is set to 1 and the IRQEn bit is set.
4	**Inhibit Switch.** This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited.
3	**CmdData.** The HC will set this bit to 0 on an I/O write to port 60h and on an I/O write to port 64h the HC will set this bit to 1.
2	**Flag.** Nominally used as a system flag by software to indicate a warm or cold boot.
1	**InputFull.** Except for the case of a Gate A20 sequence, this bit is set to 1 on an I/O write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists.
0	**OutputFull.** The HC will set this bit to 0 on a read of I/O port 60h. If IRQEn is set and AuxOutputFull is set to 0 then an IRQ1 is generated as long as this bit is set to 1. If IRQEn is set and AuxOutputFull is set to 1 then and IRQ12 will be generated a long as this bit is set to 1. While this bit is 0 and CharacterPending in HceControl is set to 1, an emulation interrupt condition exists.
Note:	This register is the emulation side of the legacy Status register.

Core Logic Module (Continued)

5.4.7 ISA Legacy Register Space

The ISA Legacy registers reside in the ISA I/O address space in the address range from 000h to FFFh and are accessed through typical input/output instructions (i.e., CPU direct R/W) with the designated I/O port address and 8-bit data.

The bit formats for the ISA Legacy I/O Registers plus two chipset-specific configuration registers used for interrupt mapping in the Core Logic module are given in this section. The ISA Legacy registers are separated into the following categories:

- DMA Channel Control Registers, see Table 5-43
- DMA Page Registers, see Table 5-44
- Programmable Interval Timer Registers, see Table 5-45
- Programmable Interrupt Controller Registers, see Table 5-46
- Keyboard Controller Registers, see Table 5-47
- Real-Time Clock Registers, see Table 5-48
- Miscellaneous Registers, see Table 5-49 (includes 4D0h and 4D1h Interrupt Edge/Level Select Registers)

Table 5-43. DMA Channel Control Registers

Bit	Description
I/O Port 000h	**DMA Channel 0 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 001h	**DMA Channel 0 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 002h	**DMA Channel 1 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 003h	**DMA Channel 1 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 004h	**DMA Channel 2 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 005h	**DMA Channel 2 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 006h	**DMA Channel 3 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 007h	**DMA Channel 3 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 008h (R/W)	
Read	**DMA Status Register, Channels 3:0**
7	**Channel 3 Request.** Indicates if a request is pending. 0: No. 1: Yes.
6	**Channel 2 Request.** Indicates if a request is pending. 0: No. 1: Yes.
5	**Channel 1 Request.** Indicates if a request is pending. 0: No. 1: Yes.
4	**Channel 0 Request.** Indicates if a request is pending. 0: No. 1: Yes.
3	**Channel 3 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
2	**Channel 2 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
1	**Channel 1 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
0	**Channel 0 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
Write	**DMA Command Register, Channels 3:0**
7	**DACK Sense.** 0: Active high. 1: Active low.
6	**DREQ Sense.** 0: Active high. 1: Active low.
5	**Write Selection.** 0: Late write. 1: Extended write.
4	**Priority Mode.** 0: Fixed. 1: Rotating.
3	**Timing Mode.** 0: Normal. 1: Compressed.
2	**Channels 3:0.** 0: Disable. 1: Enable.
1:0	**Reserved.** Must be set to 0.
I/O Port 009h	**Software DMA Request Register, Channels 3:0 (W)**
7:3	**Reserved.** Must be set to 0.
2	**Request Type.** 0: Reset. 1: Set.
1:0	**Channel Number Request Select.** 00: Channel 0. 01: Channel 1. 10: Channel 2. 11: Channel 3.
I/O Port 00Ah	**DMA Channel Mask Register, Channels 3:0 (WO)**
7:3	**Reserved.** Must be set to 0.
2	**Channel Mask.** 0: Not masked. 1: Masked.

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
1:0	**Channel Number Mask Select.** 00: Channel 0. 01: Channel 1. 10: Channel 2. 11: Channel 3.
I/O Port 00Bh	**DMA Channel Mode Register, Channels 3:0 (WO)**
7:6	**Transfer Mode.** 00: Demand. 01: Single. 10: Block. 11: Cascade.
5	**Address Direction.** 0: Increment. 1: Decrement.
4	**Auto-initialize.** 0: Disable. 1: Enable.
3:2	**Transfer Type.** 00: Verify. 01: Memory read. 10: Memory write. 11: Reserved.
1:0	**Channel Number Mode Select.** 00: Channel 0. 01: Channel 1. 10: Channel 2. 11: Channel 3.
I/O Port 00Ch	**DMA Clear Byte Pointer Command, Channels 3:0 (W)**
I/O Port 00Dh	**DMA Master Clear Command, Channels 3:0 (W)**
I/O Port 00Eh	**DMA Clear Mask Register Command, Channels 3:0 (W)**
I/O Port 00Fh	**DMA Write Mask Register Command, Channels 3:0 (W)**
I/O Port 0C0h	**DMA Channel 4 Address Register (R/W)**
Not used.	
I/O Port 0C2h	**DMA Channel 4 Transfer Count Register (R/W)**
Not used.	
I/O Port 0C4h	**DMA Channel 5 Address Register (R/W)**
Memory address bytes 1 and 0.	
I/O Port 0C6h	**DMA Channel 5 Transfer Count Register (R/W)**
Transfer count bytes 1 and 0.	
I/O Port 0C8h	**DMA Channel 6 Address Register (R/W)**
Memory address bytes 1 and 0.	
I/O Port 0CAh	**DMA Channel 6 Transfer Count Register (R/W)**
Transfer count bytes 1 and 0.	
I/O Port 0CCh	**DMA Channel 7 Address Register (R/W)**
Memory address bytes 1 and 0.	

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
I/O Port 0CEh	**DMA Channel 7 Transfer Count Register (R/W)**
Transfer count bytes 1 and 0.	
I/O Port 0D0h (R/W)	
Read	**DMA Status Register, Channels 7:4**
7	**Channel 7 Request.** Indicates if a request is pending. 0: No. 1: Yes.
6	**Channel 6 Request.** Indicates if a request is pending. 0: No. 1: Yes.
5	**Channel 5 Request.** Indicates if a request is pending. 0: No. 1: Yes.
4	**Undefined.**
3	**Channel 7 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
2	**Channel 6 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
1	**Channel 5 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
0	**Undefined.**
Write	**DMA Command Register, Channels 7:4**
7	**DACK Sense.** 0: Active high. 1: Active low.
6	**DREQ Sense.** 0: Active high. 1: Active low.
5	**Write Selection.** 0: Late write. 1: Extended write.
4	**Priority Mode.** 0: Fixed. 1: Rotating.
3	**Timing Mode.** 0: Normal. 1: Compressed.
2	**Channels 7:4.** 0: Disable. 1: Enable.
1:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
I/O Port 0D2h	**Software DMA Request Register, Channels 7:4 (W)**
7:3	**Reserved.** Must be set to 0.
2	**Request Type.** 0: Reset. 1: Set.
1:0	**Channel Number Request Select.** 00: Illegal. 01: Channel 5. 10: Channel 6. 11: Channel 7.
I/O Port 0D4h	**DMA Channel Mask Register, Channels 7:4 (WO)**
7:3	**Reserved.** Must be set to 0.
2	**Channel Mask.** 0: Not masked. 1: Masked.
1:0	**Channel Number Mask Select.** 00: Channel 4. 01: Channel 5. 10: Channel 6. 11: Channel 7.
I/O Port 0D6h	**DMA Channel Mode Register, Channels 7:4 (WO)**
7:6	**Transfer Mode.** 00: Demand. 01: Single. 10: Block. 11: Cascade.
5	**Address Direction.** 0: Increment. 1: Decrement.
4	**Auto-initialize.** 0: Disabled. 1: Enable.
3:2	**Transfer Type.** 00: Verify. 01: Memory read. 10: Memory write. 11: Reserved.
1:0	**Channel Number Mode Select.** 00: Channel 4. 01: Channel 5. 10: Channel 6. 11: Channel 7. Channel 4 must be programmed in cascade mode. This mode is not the default.
I/O Port 0D8h	**DMA Clear Byte Pointer Command, Channels 7:4 (W)**
I/O Port 0DAh	**DMA Master Clear Command, Channels 7:4 (W)**
I/O Port 0DCh	**DMA Clear Mask Register Command, Channels 7:4 (W)**
I/O Port 0DEh	**DMA Write Mask Register Command, Channels 7:4 (W)**

Core Logic Module (Continued)

Table 5-44. DMA Page Registers

Bit	Description
I/O Port 081h	**DMA Channel 2 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 082h	**DMA Channel 3 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 083h	**DMA Channel 1 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 087h	**DMA Channel 0 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 089h	**DMA Channel 6 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 08Ah	**DMA Channel 7 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 08Bh	**DMA Channel 5 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 08Fh	**ISA Refresh Low Page Register (R/W)**
Refresh address.	
I/O Port 481h	**DMA Channel 2 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 081h.	
I/O Port 482h	**DMA Channel 3 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 082h.	
I/O Port 483h	**DMA Channel 1 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 083h.	
I/O Port 487h	**DMA Channel 0 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 087h.	
I/O Port 489h	**DMA Channel 6 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 089h.	
I/O Port 48Ah	**DMA Channel 7 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 08Ah.	
I/O Port 48Bh	**DMA Channel 5 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 08Bh.	

Core Logic Module (Continued)

Table 5-45. Programmable Interval Timer Registers

Bit	Description
I/O Port 040h	
Write	**PIT Timer 0 Counter**
7:0	Counter Value.
Read	**PIT Timer 0 Status**
7	**Counter Output.** State of counter output signal.
6	**Counter Loaded.** Indicates if the last count written is loaded. 0: Yes. 1: No.
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).
I/O Port 041h	
Write	**PIT Timer 1 Counter (Refresh)**
7:0	Counter Value.
Read	**PIT Timer 1 Status (Refresh)**
7	**Counter Output.** State of counter output signal.
6	**Counter Loaded.** Indicates if the last count written is loaded. 0: Yes. 1: No.
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).

Core Logic Module (Continued)

Table 5-45. Programmable Interval Timer Registers (Continued)

Bit	Description
I/O Port 042h	
Write	**PIT Timer 2 Counter (Speaker)**
7:0	Counter Value.
Read	**PIT Timer 2 Status (Speaker)**
7	**Counter Output.** State of counter output signal.
6	**Counter Loaded.** Indicates if the last count written is loaded. 0: Yes. 1: No.
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).
I/O Port 043h (R/W)	**PIT Mode Control Word Register**
Notes: 1. If bits [7:6] = 11: Register functions as Read Status Command and: Bit 5 = Latch Count Bit 4 = Latch Status Bit 3 = Select Counter 2 Bit 2 = Select Counter 1 Bit 1 = Select Counter 0 Bit 0 = Reserved 2. If bits [5:4] = 00: Register functions as Counter Latch Command and: Bits [7:6] = Selects Counter Bits [3:0] = Don't care	
7:6	**Counter Select.** 00: Counter 0. 01: Counter 1. 10: Counter 2. 11: Read-back command (Note 1).
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).

Core Logic Module (Continued)

Table 5-46. Programmable Interrupt Controller Registers

Bit	Description
I/O Port 020h / 0A0h	**Master / Slave PIC ICW1 (WO)**
7:5	**Reserved.** Must be set to 0.
4	**Reserved.** Must be set to 1.
3	**Trigger Mode.** 0: Edge. 1: Level.
2	**Vector Address Interval.** 0: 8 byte intervals. 1: 4 byte intervals.
1	**Reserved.** Must be set to 0 (cascade mode).
0	**Reserved.** Must be set to 1 (ICW4 must be programmed).
I/O Port 021h / 0A1h	**Master / Slave PIC ICW2 (after ICW1 is written) (WO)**
7:3	**A[7:3].** Address lines [7:3] for base vector for interrupt controller.
2:0	**Reserved.** Must be set to 0.
I/O Port 021h / 0A1h	**Master / Slave PIC ICW3 (after ICW2 is written) (WO)**
Master PIC ICW3	
7:0	**Cascade IRQ.** Must be 04h.
Slave PIC ICW3	
7:0	**Slave ID.** Must be 02h.
I/O Port 021h / 0A1h	**Master / Slave PIC ICW4 (after ICW3 is written) (WO)**
7:5	**Reserved.** Must be set to 0.
4	**Special Fully Nested Mode.** 0: Disable. 1: Enable.
3:2	**Reserved.** Must be set to 0.
1	**Auto EOI.** 0: Normal EOI. 1: Auto EOI.
0	**Reserved.** Must be set to 1 (8086/8088 mode).
I/O Port 021h / 0A1h (R/W)	**Master / Slave PIC OCW1 (except immediately after ICW1 is written)**
7	**IRQ7 / IRQ15 Mask.** 0: Not Masked. 1: Mask.
6	**IRQ6 / IRQ14 Mask.** 0: Not Masked. 1: Mask.
5	**IRQ5 / IRQ13 Mask.** 0: Not Masked. 1: Mask.
4	**IRQ4 / IRQ12 Mask.** 0: Not Masked. 1: Mask.
3	**IRQ3 / IRQ11 Mask.** 0: Not Masked. 1: Mask.

Core Logic Module (Continued)

Table 5-46. Programmable Interrupt Controller Registers (Continued)

Bit	Description
2	**IRQ2 / IRQ10 Mask.** 0: Not Masked. 1: Mask.
1	**IRQ1 / IRQ9 Mask.** 0: Not Masked. 1: Mask.
0	**IRQ0 / IRQ8 Mask.** 0: Not Masked. 1: Mask.
I/O Port 020h / 0A0h	**Master / Slave PIC OCW2 (WO)**
7:5	**Rotate/EOI Codes.** 000: Clear rotate in Auto EOI mode 001: Non-specific EOI 010: No operation 011: Specific EOI (bits [2:0] must be valid) 100: Set rotate in Auto EOI mode 101: Rotate on non-specific EOI command 110: Set priority command (bits [2:0] must be valid) 111: Rotate on specific EOI command
4:3	**Reserved.** Must be set to 0.
2:0	**IRQ Number (000-111)**
I/O Port 020h / 0A0h	**Master / Slave PIC OCW3 (WO)**
7	**Reserved.** Must be set to 0.
6:5	**Special Mask Mode.** 00: No operation. 01: No operation. 10: Reset Special Mask Mode. 11: Set Special Mask Mode.
4	**Reserved.** Must be set to 0.
3	**Reserved.** Must be set to 1.
2	**Poll Command.** 0: Disable. 1: Enable.
1:0	**Register Read Mode.** 00: No operation. 01: No operation. 10: Read interrupt request register on next read of Port 20h. 11: Read interrupt service register on next read of Port 20h.
I/O Port 020h / 0A0h	**Master / Slave PIC Interrupt Request and Service Registers for OCW3 Commands (RO)**
The function of this register is set with bits [1:0] in a write to 020h.	
Interrupt Request Register	
7	**IRQ7 / IRQ15 Pending.** 0: Yes. 1: No.
6	**IRQ6 / IRQ14 Pending.** 0: Yes. 1: No.
5	**IRQ5 / IRQ13 Pending.** 0: Yes. 1: No.

Core Logic Module (Continued)

Table 5-46. Programmable Interrupt Controller Registers (Continued)

Bit	Description
4	**IRQ4 / IRQ12 Pending.** 0: Yes. 1: No.
3	**IRQ3 / IRQ11 Pending.** 0: Yes. 1: No.
2	**IRQ2 / IRQ10 Pending.** 0: Yes. 1: No.
1	**IRQ1 / IRQ9 Pending.** 0: Yes. 1: No.
0	**IRQ0 / IRQ8 Pending.** 0: Yes. 1: No.
Interrupt Service Register	
7	**IRQ7 / IRQ15 In-Service.** 0: No. 1: Yes.
6	**IRQ6 / IRQ14 In-Service.** 0: No. 1: Yes.
5	**IRQ5 / IRQ13 In-Service.** 0: No. 1: Yes.
4	**IRQ4 / IRQ12 In-Service.** 0: No. 1: Yes.
3	**IRQ3 / IRQ11 In-Service.** 0: No. 1: Yes.
2	**IRQ2 / IRQ10 In-Service.** 0: No. 1: Yes.
1	**IRQ1 / IRQ9 In-Service.** 0: No. 1: Yes.
0	**IRQ0 / IRQ8 In-Service.** 0: No. 1: Yes.

Core Logic Module (Continued)

Table 5-47. Keyboard Controller Registers

Bit	Description
I/O Port 060h	**External Keyboard Controller Data Register (R/W)**
	Keyboard Controller Data Register. All accesses to this port are passed to the ISA bus. If the fast keyboard gate A20 and reset features are enabled through bit 7 of the ROM/AT Logic Control Register (F0 Index 52h[7]), the respective sequences of writes to this port assert the A20M# signal or cause a warm CPU reset.
I/O Port 061h	**Port B Control Register (R/W)** **Reset Value: 00x01100b**
7	**PERR#/SERR# Status. (Read Only)** Indicates if a PCI bus error (PERR#/SERR#) was asserted by a PCI device or by the SC2200. 0: No. 1: Yes. This bit can only be set if ERR_EN (bit 2) is set 0. This bit is set 0 after a write to ERR_EN with a 1 or after reset.
6	**IOCHK# Status. (Read Only)** Indicates if an I/O device is reporting an error to the SC2200. 0: No. 1: Yes. This bit can only be set if IOCHK_EN (bit 3) is set 0. This bit is set 0 after a write to IOCHK_EN with a 1 or after reset.
5	**PIT OUT2 State. (Read Only)** This bit reflects the current status of the of the PIT Counter 2 (OUT2).
4	**Toggle. (Read Only)** This bit toggles on every falling edge of Counter 1 (OUT1).
3	**IOCHK# Enable** 0: Generates an NMI if IOCHK# is driven low by an I/O device to report an error. Note that NMI is under SMI control. 1: Ignores the IOCHK# input signal and does not generate NMI.
2	**PERR/ SERR Enable.** Generate an NMI if PERR#/SERR# is driven active to report an error. 0: Enable. 1: Disable.
1	**PIT Counter2 (SPKR)** 0: Forces Counter 2 output (OUT2) to zero. 1: Allows Counter 2 output (OUT2) to pass to the speaker.
0	**PIT Counter2 Enable.** 0: Sets GATE2 input low. 1: Sets GATE2 input high.
I/O Port 062h	**External Keyboard Controller Mailbox Register (R/W)**
	Keyboard Controller Mailbox Register.
I/O Port 064h	**External Keyboard Controller Command Register (R/W)**
	Keyboard Controller Command Register. All accesses to this port are passed to the ISA bus. If the fast keyboard gate A20 and reset features are enabled through bit 7 of the ROM/AT Logic Control Register (F0 Index 52h[7]), the respective sequences of writes to this port assert the A20M# signal or cause a warm CPU reset.
I/O Port 066h	**External Keyboard Controller Mailbox Register (R/W)**
	Keyboard Controller Mailbox Register.
I/O Port 092h	**Port A Control Register (R/W)** **Reset Value: 02h**
7:2	**Reserved.** Must be set to 0.
1	**A20M# Assertion.** Assert A20# (internally). 0: Enable. 1: Disable. This bit reflects A20# status and can be changed by keyboard command monitoring. An SMI event is generated when this bit is changed, if enabled by F0 Index 53h[0]. The SMI status is reported in F1BAR0+I/O Offset 00h/02h[7].
0	**Fast CPU Reset.** WM_RST SMI is asserted to the BIOS. 0: Disable. 1: Enable. This bit must be cleared before the generation of another reset.

Core Logic Module (Continued)

Table 5-48. Real-Time Clock Registers

Bit	Description
I/O Port 070h	**RTC Address Register (WO)**
This register is shadowed within the Core Logic module and is read through the RTC Shadow Register (F0 Index BBh).	
7	**NMI Mask**. 0: Enable. 1: Mask.
6:0	**RTC Register Index.** A write of this register sends the data out on the ISA bus and also causes RTCALE to be triggered. (RTCALE is an internal signal between the Core Logic module and the internal RTC controller.)
I/O Port 071h	**RTC Data Register (R/W)**
A read of this register returns the value of the register indexed by the RTC Address Register. A write of this register sets the value into the register indexed by the RTC Address Register.	
I/O Port 072h	**RTC Extended Address Register (WO)**
7	**Reserved.**
6:0	**RTC Register Index.** A write of this register sends the data out on the ISA bus and also causes RTCALE to be triggered. (RTCALE is an internal signal between the Core Logic module and the internal RTC controller.)
I/O Port 073h	**RTC Data Register (R/W)**
AA read of this register returns the value of the register indexed by the RTC Extended Address Register. A write of this register sets the value into the register indexed by the RTC Extended Address Register.	

Table 5-49. Miscellaneous Registers

Bit	Description
I/O Port 0F0h, 0F1h	**Coprocessor Error Register (W)** **Reset Value: F0h**
A write to either port when the internal FERR# signal is asserted causes the Core Logic Module to assert internal IGNNE#. IGNNE# remains asserted until the FERR# de-asserts.	
I/O Ports 170h-177h/376h-377h	**Secondary IDE Registers (R/W)**
When the local IDE functions are enabled, reads or writes to these registers cause the local IDE interface signals to operate according to their configuration rather than generating standard ISA bus cycles.	
I/O Ports 1F0h-1F7h/3F6h-3F7h	**Primary IDE Registers (R/W)**
When the local IDE functions are enabled, reads or writes to these registers cause the local IDE interface signals to operate according to their configuration rather than generating standard ISA bus cycles.	
I/O Port 4D0h	**Interrupt Edge/Level Select Register 1 (R/W)** **Reset Value: 00h**
Notes: 1. If ICW1 - bit 3 in the PIC is set as level, it overrides the setting for bits [7:3] in this register. 2. Bits [7:3] in this register are used to configure a PCI interrupt mapped to IRQ[x] on the PIC as level-sensitive (shared).	
7	**IRQ7 Edge or Level Sensitive Select.** Selects PIC IRQ7 sensitivity configuration. 0: Edge. 1: Level.
6	**IRQ6 Edge or Level Sensitive Select.** Selects PIC IRQ6 sensitivity configuration. 0: Edge. 1: Level.
5	**IRQ5 Edge or Level Sensitive Select.** Selects PIC IRQ5 sensitivity configuration. 0: Edge. 1: Level.
4	**IRQ4 Edge or Level Sensitive Select.** Selects PIC IRQ4 sensitivity configuration. 0: Edge. 1: Level.

Core Logic Module (Continued)

Table 5-49. Miscellaneous Registers (Continued)

Bit	Description
3	**IRQ3 Edge or Level Sensitive Select.** Selects PIC IRQ3 sensitivity configuration. 0: Edge. 1: Level.
2:0	**Reserved**. Must be set to 0.
I/O Port 4D1h	**Interrupt Edge/Level Select Register 2 (R/W)** **Reset Value: 00h**
Notes: 1. If ICW1 - bit 3 in the PIC is set as level, it overrides the setting for bits 7:6 and 4:1 in this register. 2. Bits [7:6] and [4:1] in this register are used to configure a PCI interrupt mapped to IRQ[x] on the PIC as level-sensitive (shared).	
7	**IRQ15 Edge or Level Sensitive Select.** Selects PIC IRQ15 sensitivity configuration. 0: Edge. 1: Level.
6	**IRQ14 Edge or Level Sensitive Select.** Selects PIC IRQ14 sensitivity configuration. 0: Edge. 1: Level.
5	**Reserved**. Must be set to 0.
4	**IRQ12 Edge or Level Sensitive Select.** Selects PIC IRQ12 sensitivity configuration. 0: Edge. 1: Level.
3	**IRQ11 Edge or Level Sensitive Select.** Selects PIC IRQ11 sensitivity configuration. 0: Edge. 1: Level.
2	**IRQ10 Edge or Level Sensitive Select.** Selects PIC IRQ10 sensitivity configuration. 0: Edge. 1: Level.
1	**IRQ9 Edge or Level Sensitive Select.** Selects PIC IRQ9 sensitivity configuration. 0: Edge. 1: Level.
0	**Reserved.** Must be set to 0.

6.0 Video Processor Module

The Video Processor module contains a high performance video back-end accelerator, a video/graphics Mixer/Blender, and a Video Input Port (VIP), supporting two output choices: CRT or TFT. The back-end accelerator functions include horizontal and vertical scaling and filtering of the video stream. The Mixer/Blender function includes color space conversion, gamma correction, and mixing or alpha blending the video and graphics streams.

General Features

- Hardware video acceleration
- Graphics/video overlay and blending
- Integrated CRT DACs and PLL
- Selection of interlaced and progressive video from the GX1 module and the Direct Video Port

Video Input Port (VIP)

- CCIR-656 compatible
- Capture Video/VBI modes
- Direct Video/VBI modes

Hardware Video Acceleration

- Arbitrary X and Y interpolation using three line-buffers
- YUV-to-RGB color space conversion
- Horizontal filtering and downscaling
- Supports 4:2:2, 4:2:0 YUV formats and RGB 5:6:5 format

Graphics-Video Overlay and Blending

- Overlay of video up to 16 bpp
- Supports chroma key and color key for both graphics and video streams
- Supports alpha-blending with up to three alpha windows that can overlap one another
- 8-Bit alpha values with automatic increment or decrement on each frame
- Optional Gamma Correction for video or graphics

Compatibility

- Supports Microsoft's DirectDraw/Direct Video and Display Control Interface (DCI) Version 2.0 for full motion playback acceleration
- Compliant with PC98 and PC99 V0.7
- Compatible with VESA, VGA, DPMS, and DDC2 standards for enhanced display control and power management

Integrated CRT DACs and PLL

- Support up to 135 MHz (three 8-bit DACs)
- PLL rate up to 135 MHz

Display Modes

- Supported CRT modes:
 - 640x480x16 bpp at 60-85 Hz vertical refresh rates
 - 800x600x16 bpp at 60-85 Hz vertical refresh rates
 - 1024x768x16 bpp at 60-85 Hz vertical refresh rates
 - 1280x1024x8 bpp at 60-75 Hz vertical refresh rates
- Supported TFT modes:
 - TFT on IDE: FPCLK max is 40 MHz
 - TFT on Parallel Port: FPCLK max is 80 MHz
 - 640x480x16 bpp at 60-85 Hz vertical refresh rates
 - 800x600x16 bpp at 60-85 Hz vertical refresh rates
 - 1024x768x16 bpp at 60-75 Hz vertical refresh rates
 - 1280x1024x8 bpp at 60 Hz vertical refresh rate

Video Processor Module (Continued)

6.1 MODULE ARCHITECTURE

Figure 6-1 shows a top-level block diagram of the Video Processor. For information about the relationship between the Video Processor and the other modules of the SC2200, see Section 1.2 on page 16. The Video Processor module includes the following functions:

- Video Input Port
 - — CCIR-656 decoder
 - — Capture Video/VBI modes
 - — Direct Video mode
- Video Formatter
 - — Asynchronous video interface
 - — Horizontal/vertical scalers
 - — Filters
- Mixer/Blender
 - — Overlay with color/chroma key
 - — Gamma correction
 - — Color space converters
 - — Alpha blender
- Outputs
 - — CRT interface with DACs
 - — TFT interface
- Dot Clock PLL

The following subsections describe each block in detail.

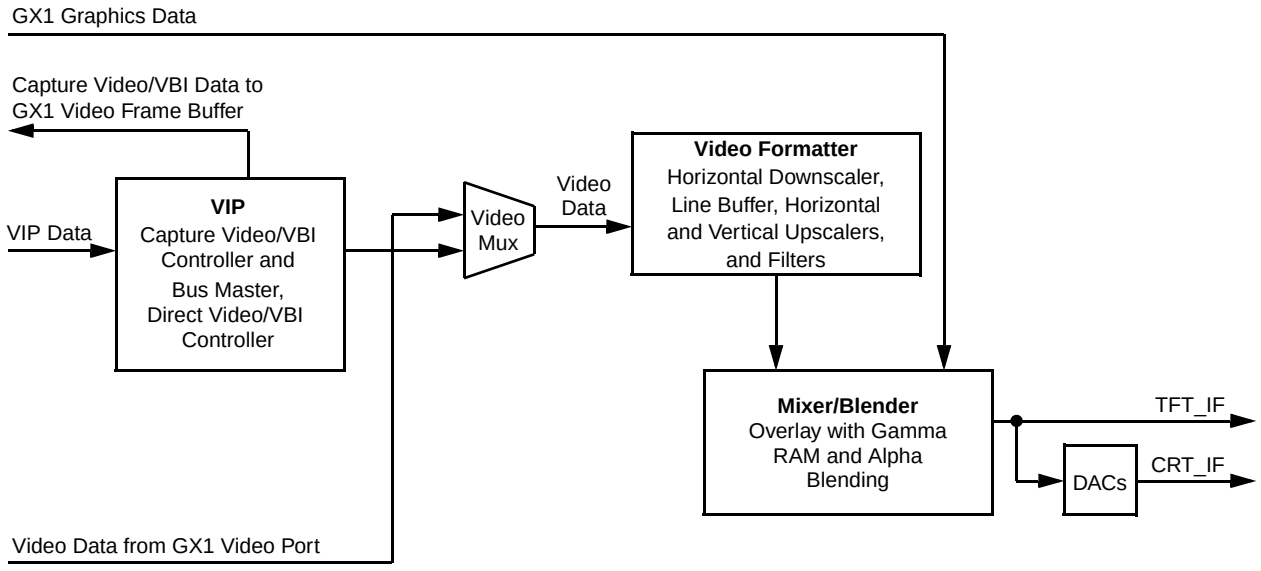


Figure 6-1. Video Processor Block Diagram

Video Processor Module (Continued)

6.2 FUNCTIONAL DESCRIPTION

To understand why the Video Processor functions as it does, it is first important to understand the difference between video and graphics. Video is pictures in motion, which usually starts out in an encoded format (i.e., MPEG2, AVI, MPEG4) or is a TV broadcast. These pictures or frames are generally dynamic and are drawn 24 to 30 frames per second. Conversely, graphic data is relatively static and is drawn - usually using hardware accelerators. Most IA devices need to support both video and graphics displayed at the same time. For some IA devices, such as set-top boxes, video is dominant. While for other devices, such as consumer access devices and thin clients, graphics is dominant. What this means for the Video Processor is that for video centric devices, graphics overlays the video; and for graphics centric devices, video overlays the graphics.

Video Support

The SC2200 gets video from two sources, either the VIP block or the GX1 module's video frame buffer. The VIP block supports the CCIR-656 data protocol. The CCIR-656 protocol supports TV data (NTSC or PAL) and defines the format for active video data and vertical blanking interval (VBI) data. Conforming CCIR-656 data matches exactly what is needed for a TV: full frame, interlaced, 27 MHz pixel clock, and 50 or 60 Hz refresh rate. Full frame pixel resolution and the refresh rate depends on the TV standard: NTSC, PAL, or SECAM.

If the VIP input data is full frame (conforming data), the data can go directly from the VIP block to the Video Formatter. This is known as Direct Video mode. In this mode, the data never leaves the Video Processor module. Direct Video mode can only be used under very specific conditions which will be explained later. If the VIP data is less than full frame (non conforming data), the VIP block will bus master the video data to the GX1 module's video frame buffer. The GX1 module's display controller then moves the video data out of the video frame buffer and sends it to the Video Formatter. Using this method the temporal (refresh rate) and/or spatial (image less then full screen) differences between the VIP data and the output device are reconciled. This method is known as Capture Video mode. How each mode is setup and operates is explained further in Section 6.2.1 on page 320.

VBI Support

VBI (vertical blanking interval) data is placed in the video data stream during a portion of the vertical retrace period. The vertical retrace period physically consists of several horizontal lines (24 for NTSC and 25 for PAL systems) of non-active video. Data can be placed on some of these lines for other uses.

The active video and vertical retrace period horizontal lines are logically defined into 23 types: logical line 2 through logical line 24 (no logical line 1). Logical lines 2 through 23 occur during the vertical retrace period and logical line 24 represents all the active video lines. Logical lines 10 through 21 for NTSC and 6 through 23 for PAL are the nominal VBI lines. The rest of the logical lines, 2 through 9, 22, and 23 for NTSC and 2 through 6 for PAL occur during the vertical retrace period but do not normally carry user data. An example of VBI usage is Closed Captioning, which occupies VBI logical line 21 for NTSC. Figure 6-2 and Figure 6-3 on page 319 show the (relationship between the) physical scan lines and logical scan lines for the odd and even fields in the NTSC format.

Video Processor Module (Continued)

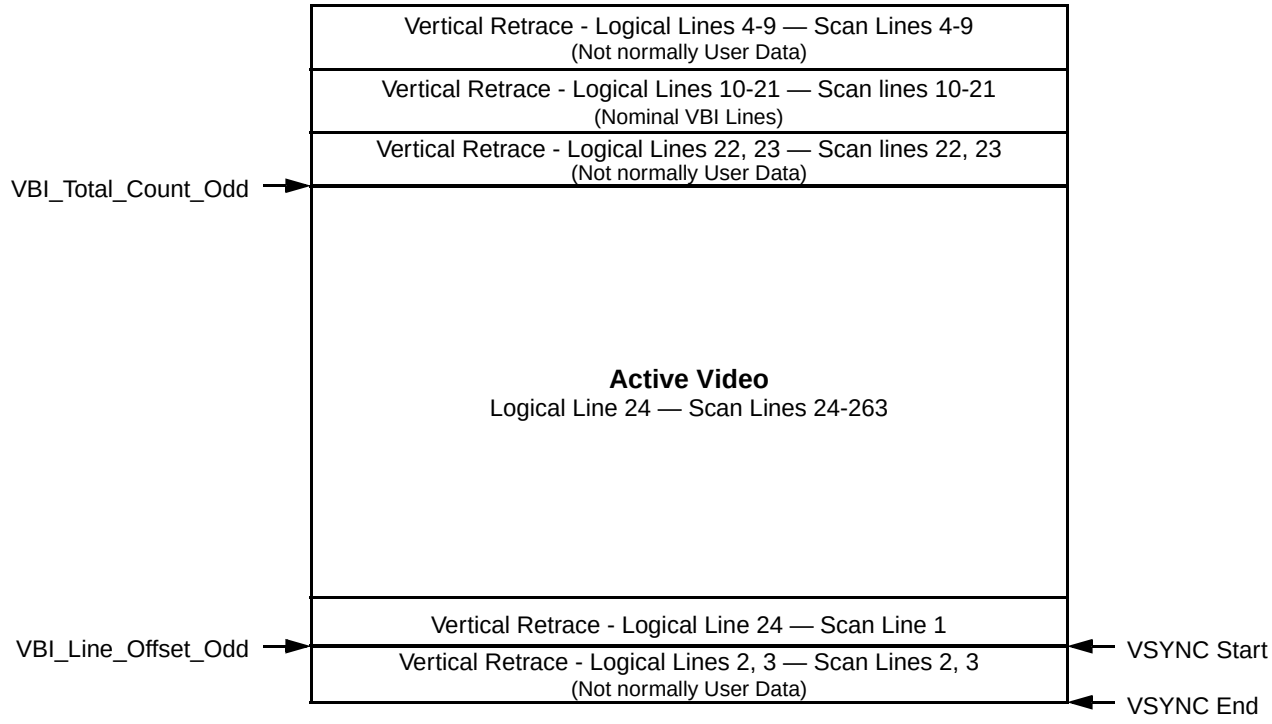


Figure 6-2. NTSC 525 Lines, 60 Hz, Odd Field

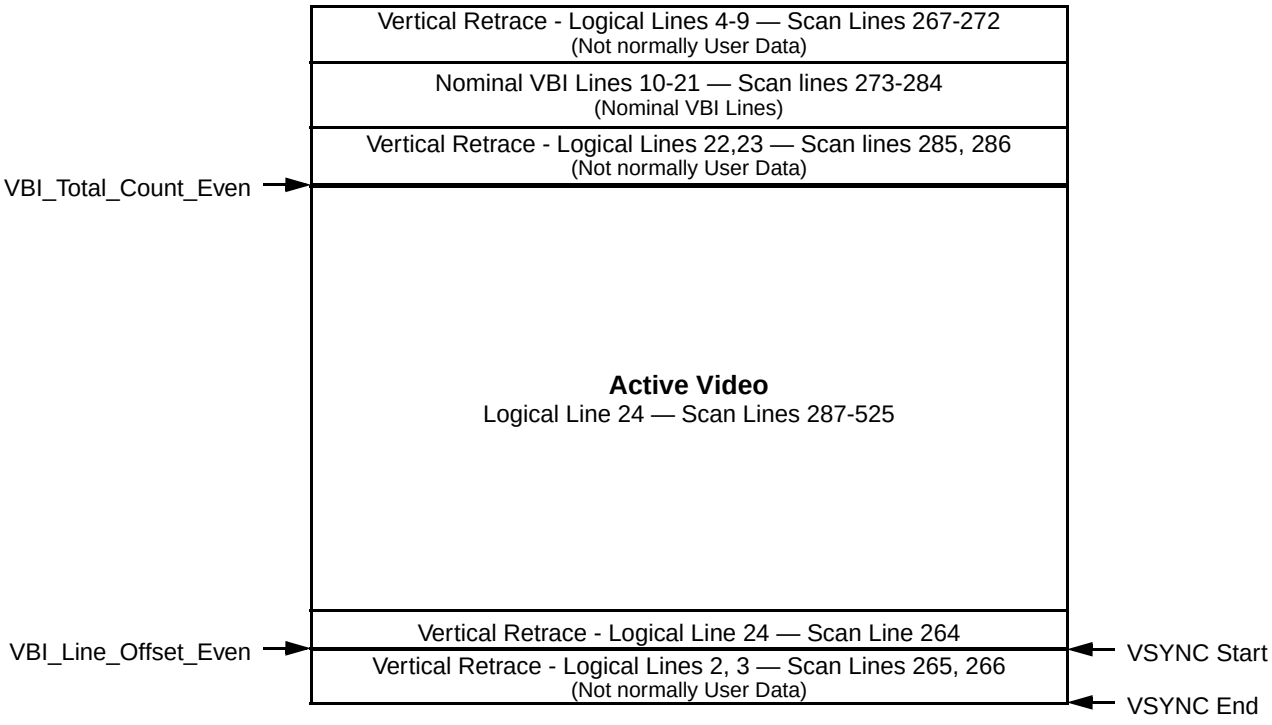


Figure 6-3. NTSC 525 Lines, 60 Hz, Even Field

Video Processor Module (Continued)

6.2.1 Video Input Port (VIP)

The VIP block is designed to interface the SC2200 with external video processors (e.g., Geode CS1300 or Sigma Designs EM8400) or external TV decoders (e.g., Philips SAA7114). It inputs CCIR-656 Video and VBI data sourced by those devices, decodes the data, and delivers the data directly to the Video Formatter (Direct Video mode) or to the GX1 module's video frame buffer (Capture Video/VBI modes). Figure 6-4 shows a diagram of the VIP block.

From the VIP block's perspective, Direct Video mode is always on. There are no registers that enable/disable Direct Video mode. The data source selected at the video mux (F4BAR0+Memory Offset 400h[1:0]) determines if the data from the VIP interface is moved directly or must be captured.

Two FIFOs in the VIP block support the efficient movement of Video and VBI data. For Capture Video/VBI modes, a 128-byte FIFO buffers both Video and VBI data processed by the CCIR-656 decoder. For Direct Video mode, there is a 2048-byte FIFO that buffer the CCIR-656 decoder's video data. The FIFOs are also used to provide clock domain changes. The VIP interface clock (nominally 27 MHz) is the input clock domain for both FIFOs. For the Capture Video/VBI FIFO, the data is clocked out using the FPCI clock (33 or 66 MHz). For the Direct Video FIFO, the Video data is clocked out using the GX1's Video port clock (75, 116, or 133 MHz GX1 core clock divided by 2 or 4).

6.2.1.1 Direct Video Mode

As stated previously, Direct Video mode is on by default so no registers need to be programmed to support this mode other than to select the direct video data at the video mux. The video mux control register is located at F4BAR0+Memory Offset 400h[1:0].

Direct Video mode while supported is not an optimal mode of operation. The vertical sync signal is affected by this mode which cause some CRT monitors operate incorrectly. This mode supports only one vertical resolution and refresh rate, which is that of the incoming data. Horizontal resolution can be scaled if desired. Since the incoming data has odd and even fields, incoming line must be doubled for it to display properly. This is equivalent to the Bob technique which is explained later in this section.

GenLock

Because video input data from the VIP is sent directly, without significant buffering frame-to-field synchronization is required with the GX1 module's graphics data. This synchronization is known as GenLock. The GenLock registers are located at F4BAR0+Memory Offset 420h and 424h.

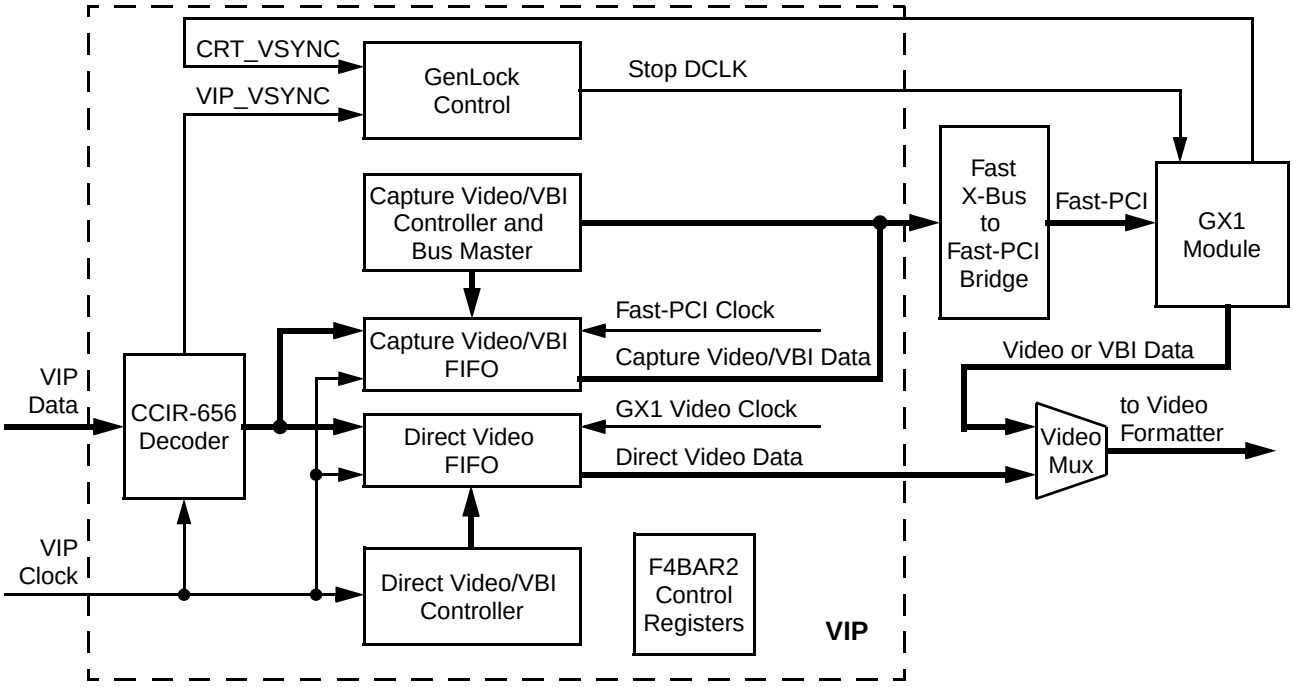


Figure 6-4. VIP Block Diagram

The GenLock control hardware is used to synchronize the video input's field with the GX1 module's graphics frame. The graphics data is always sent full frame. For the GenLock function to perform correctly, the GX1 module's Display Controller must be programmed to have a slightly faster frame time then the video input's field time. This is best accomplished by programming the GX1 module's Display Controller with a few less (three to five) horizontal lines then the VIP interface. GenLock is accomplished by stopping the clock driving the GX1 module's graphics frame until the VIP vertical sync occurs (plus some additional delay, via F4BAR0+Memory Offset 424h).

The GenLock function provides a timeout feature (GENLOCK_TOUT_EN, F4BAR0+Memory Offset 420h[4]) in case the video port input clock stops due to a problem with incoming video.

6.2.1.2 Capture Video Mode

Capture Video mode is a process for bus mastering Video data received from the VIP block to the GX1 module's Video Frame Buffer. The GX1 module's Display Controller then moves the data from the Video Frame Buffer to the Video Formatter. Usually Capture Video mode is used because the data coming in from the VIP block is interlaced and has a 30 Hz refresh rate (NTSC format) and the output device, CRT monitor or TFT panel, is progressive and has a 60 to 85 Hz refresh rate. The Capture Video mode process must convert the interlaced data to progressive data and change the frames per second. There are two methods to perform the interlaced to progressive conversion; Bob and Weave. Each method uses a different mechanism to up the refresh rate

Bob

The Bob method displays the odd frame followed by the even frame. If a full-scale image is displayed, each line in the odd and even field must be vertically doubled (see Section 6.2.2.5 "2-Tap Vertical and Horizontal Upscalers" on page 326) because each odd and each even field only contain one-half a frames worth of data. This means that the Bob method reduces the video image resolution, but has a higher effective refresh rate. If there is a change of refresh rate from the VIP block to the display device, then a field will sometimes be displayed twice. The advantage of this method is that the process is simple as only half the data is transmitted from the GX1 module's Video Frame Buffer to the Video Processor per a given amount of time, therefore reducing the memory bandwidth requirement. The disadvantage is that there are some observable visual effects due to the reduction in resolution.

Figure 6-5 on page 322 is an example of how the Bob method is performed. The example assumes that the display device is a CRT at 85 Hz refresh and single buffering is used for the data. The example does not assume anything regarding scaling that may be performed in the Video Processor. The example is only presented to allow for a general understanding of how the SC2200's video support hardware works and not as an all-inclusive statement of operation.

The following procedure is an example of how to create a Bob method. This example assumes single buffering in the GX1 module's video frame buffer. The Video Processor registers that control the VIP bus master only need to be initialized.

1) **Program the VIP bus master address registers.**
 Three registers control where the VIP video data is stored in the GX1 module's frame buffer:
 - F4BAR2+Memory Offset 20h – Video Data Odd Base Address
 - F4BAR2+Memory Offset 24h – Video Data Even Base Address
 - F4BAR2+Memory Offset 28h – Video Data Pitch

 The Video Data Even Base Address must be separated from the Video Data Odd Base Address by at least the field data size. The Video Data Pitch register must be programmed to 00000000h.

2) **Program other VIP bus master support registers.**
 In F4BAR2+Memory Offset 00h, make sure that the VIP FIFO bus request threshold is set to 32 bytes (bit 22 = 1) and that the Video Input Port mode is set to CCIR-656. An interrupt needs to be generated so that the GX1 module's video frame buffer pointer can flip to the field that has completed transfer to the video frame buffer. So in F4BAR2+Memory Offset 04h, enable the Field Interrupt bit. Auto-Flip is normally set to allow the CCIR-656 Decoder to identify which field is being processed. Capture video data needs to be enabled and Run Mode Capture is set to Start Capture at beginning of next field. Data is now being captured to the frame buffer.

3) **Field Interrupt.**
 When the field interrupt occurs, the interrupt handler must program the GX1 module's video buffer start offset value (GX_BASE+Memory Offset 8320h) with the address of the field that was just received from the VIP interface. This action will cause the display controller to ping-pong between the two fields. The new address will not take affect until the start of a new display controller frame. The field that was just received can be known by reading the Current Field bit at F4BAR2+Memory Offset 08h[24].

Video Processor Module (Continued)

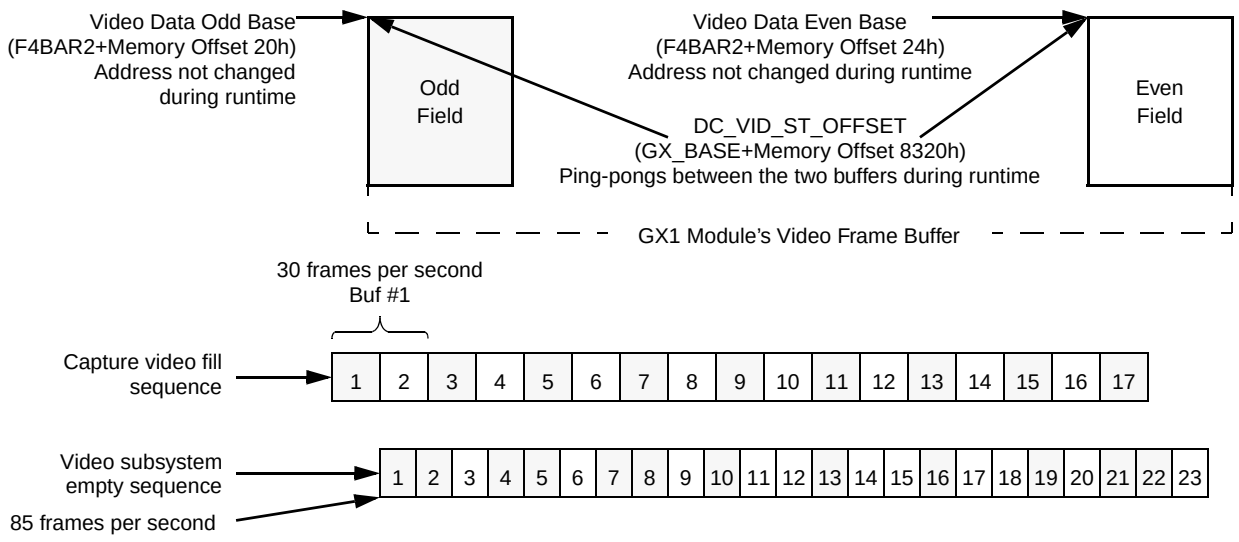


Figure 6-5. Capture Video Mode Bob Example Using One Video Frame Buffer

Weave

The Weave method assembles the odd field and even field together to form the complete frame, and then renders the "weaved" frames to the display device. The Video data is converted from interlaced to progressive. Since both fields are rendered simultaneously, the GX1 module's video frame buffer must be at least double buffered. The Weave method has the advantage of not creating the temporal effects that Bob does. The disadvantage of Weave is twice as much data is transferred from the video frame buffer to the Video Processor; meaning that Weave uses more memory bandwidth.

Figure 6-6 on page 323 is an example of the Weave method in action. As in the Bob example (Figure 6-5), a CRT monitor at 85 Hz refresh is assumed. Double buffering of the incoming data is also assumed. The example does not assume anything about any scaling that may be done in the Video Processor. No attempt has been made to assure that this example is absolutely workable. The example is only presented to allow for a general understanding of how the SC2200's video support hardware works.

The following procedure is an example of how to create the Weave method. Since at least double buffering is required, more of the VIP's control registers are used for Weave than required for Bob during video runtime.

1) **Program the VIP bus master address registers.**
Three registers control where the VIP video data is stored in the GX1 module's frame buffer:

 – F4BAR2+Memory Offset 20h – Video Data Odd Base Address

 – F4BAR2+Memory Offset 24h – Video Data Even Base Address

 – F4BAR2+Memory Offset 28h – Video Data Pitch

 The Video Data Even Base Address must be separated from the Video Data Odd Base Address by one horizontal line. The Video Data Pitch register must be programmed to one horizontal line.

2) **Program other VIP bus master support registers.**
Ensure the VIP FIFO Bus Request Threshold is set to 32 bytes (F4BAR2+Memory Offset 00h[22] = 1) and the Video Input Port mode is set to CCIR-656 (F4BAR2+Memory Offset 00h[1:0] = 10). An interrupt needs to be generated so that the GX1 module's video frame buffer pointer can flip to the field that has completed transfer to the video frame buffer. So the Field Interrupt bit (F4BAR2+Memory Offset 04h[16] = 1). must be enabled. Auto-Flip is normally set (F4BAR2+Memory Offset 04h[10] = 0) to allow the CCIR-656 decoder to identify which field is being processed. Capture video data needs to be enabled (F4BAR2+Memory Offset 04h[10] = 1) and Run Mode Capture is set to Start Capture (F4BAR2+Memory Offset 04h[1:0] = 11) at beginning of next field. Data is now being captured to the frame buffer.

Video Processor Module (Continued)

3) Field Interrupt.
When the field interrupt occurs on the completion of an odd field, the interrupt must program the Video Data Odd Base Address with the other buffer's address. The odd field will ping-pong between the two buffers. When the interrupt is due to the completion of an even field, the interrupt handler must program the GX1 module's video buffer start offset value (GX_BASE+Memory Offset 8320h) with the address of the frame (both odd and even fields) that was just received from the VIP block. This new address will not take affect until the start of a new frame. It must also program the Video Data Even Base Address with the other buffer so that the even field will ping-pong just like the odd field. The field just received can be known by reading the Current Field bit (F4BAR2+Memory Offset 08h[24]).

6.2.1.3 Capture VBI Mode

There are three types of VBI data defined by the CCIR-656 protocol: Task A data, Task B data, and Ancillary data. The VIP block supports the capture for each data type. Generally Task A data is the data type captured. Just as in Capture Video mode, there are three registers that tell the bus master where to put the VBI data in the GX1 module's frame buffer. Once the VBI data has been captured, the data can be manipulated or decoded. The data can also be used by an application. An example of this would be an Internet address that is encoded on one or more of the VBI lines, or have an application decode the Closed Captioning information put in the graphics frame buffer.

The registers, F4BAR2+Memory Offset 40h, 44h, and 48h, tell the bus master the destination addresses for the VBI data in the GX1 module's frame buffer. Five bits (F4BAR2+Memory Offset 00h[21:17]) are used to tell the bus master the data types to store. Capture VBI mode needs to be enabled at F4BAR2+Memory Offset 04h[9,1:0]. The Field Interrupt bit (F4BAR2+Memory Offset 04h[16]) should be used by the software driver to know when the captured VBI data has been completed for a field.

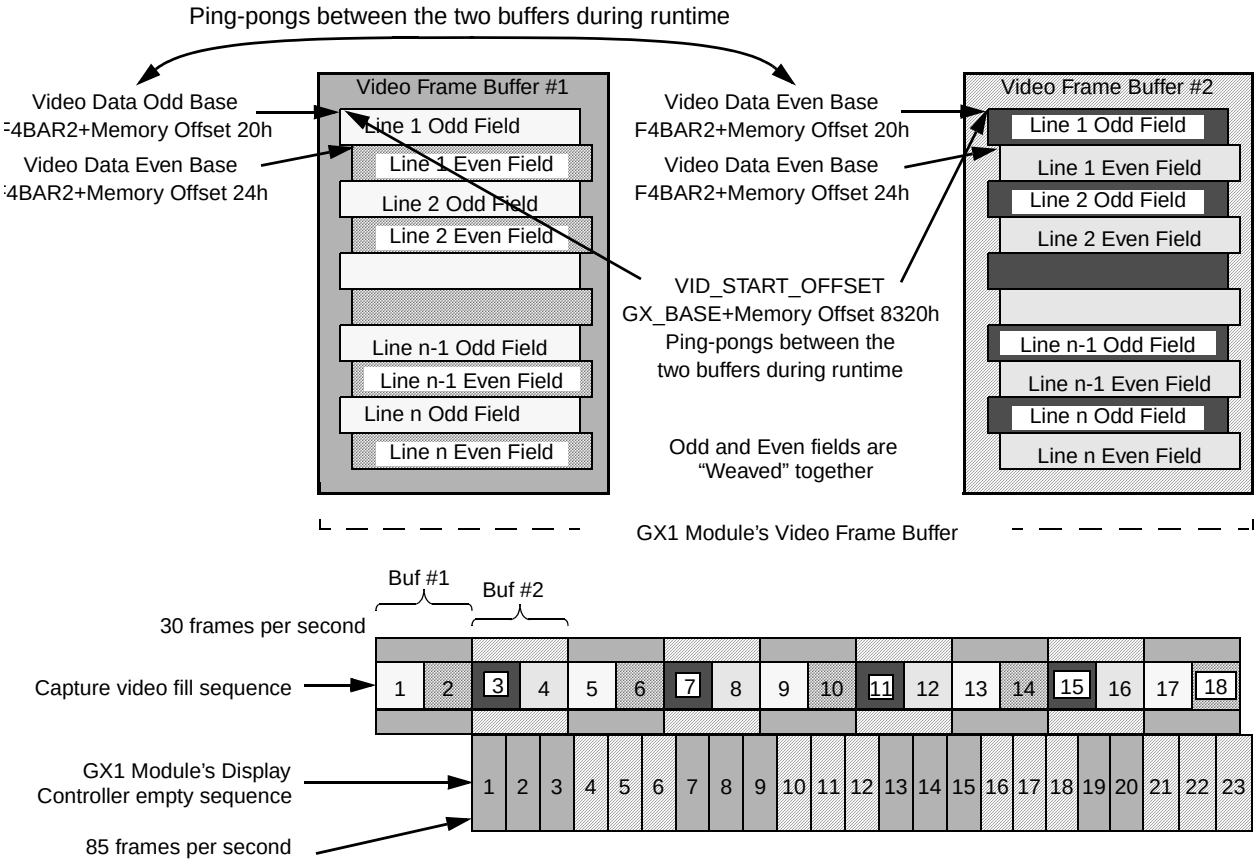


Figure 6-6. Capture Video Mode Weave Example Using Two Video Frame Buffers

Video Processor Module (Continued)

6.2.2 Video Block

The Video block receives video data from the VIP block or the GX1 module's video frame buffer. The video data is formatted and scaled and then sent to the Mixer/Blender. The video data also changes clock domains while in the Video block. It is clocked in with the GX1 module's video clock and it is clocked out with the GX1 module's graphics clock. A diagram of the Video block is shown in Figure 6-7.

6.2.2.1 Video Input Formatter

The Video Input Formatter accepts video data 8 bits at a time in YUV 4:2:2, YUV 4:2:0, or RGB 6:5:6 format. The GX1 module's video clock is the source clock. The data can be interlaced or progressive. When the data comes directly from the VIP block it is usually interlaced. The video format is configured via the EN_42X bit (F4BAR0+Memory Offset 00h[28] and the GV_SEL bit (F4BAR0+Memory Offset 4Ch[13]). The byte order for each format is configured in the VID_FMT bits (F4BAR0+Offset 00h[3:2]).

RGB 5:6:5 – For this format each pixel is described as a 16-bit value:

Bits [15:11] = Red
Bits [10:5] = Green
Bits [4:0] = Blue

YUV 4:2:0 – This format is not supported by the GX1 module. The Horizontal Downscaler in the Video block cannot be used if the video data is in this format. In this format, 4 bytes of data are used to describe two pixels. The 4 bytes contain two Y values one for each pixel; one U and one V for both pixels. For each horizontal line, all the Y values are received first. The U values are received next and the V values are received last. For example for a horizontal line that has 720 pixels, there are 720 bytes of Y, followed by 360 bytes of U, followed by 360 bytes of V.

YUV 4:2:2 – In this format each DWORD in the horizontal line represent two pixels. There are two Y values and one each U and V in a DWORD. Just as in the YUV 4:2:0 format, each U and V value describes the two pixels.

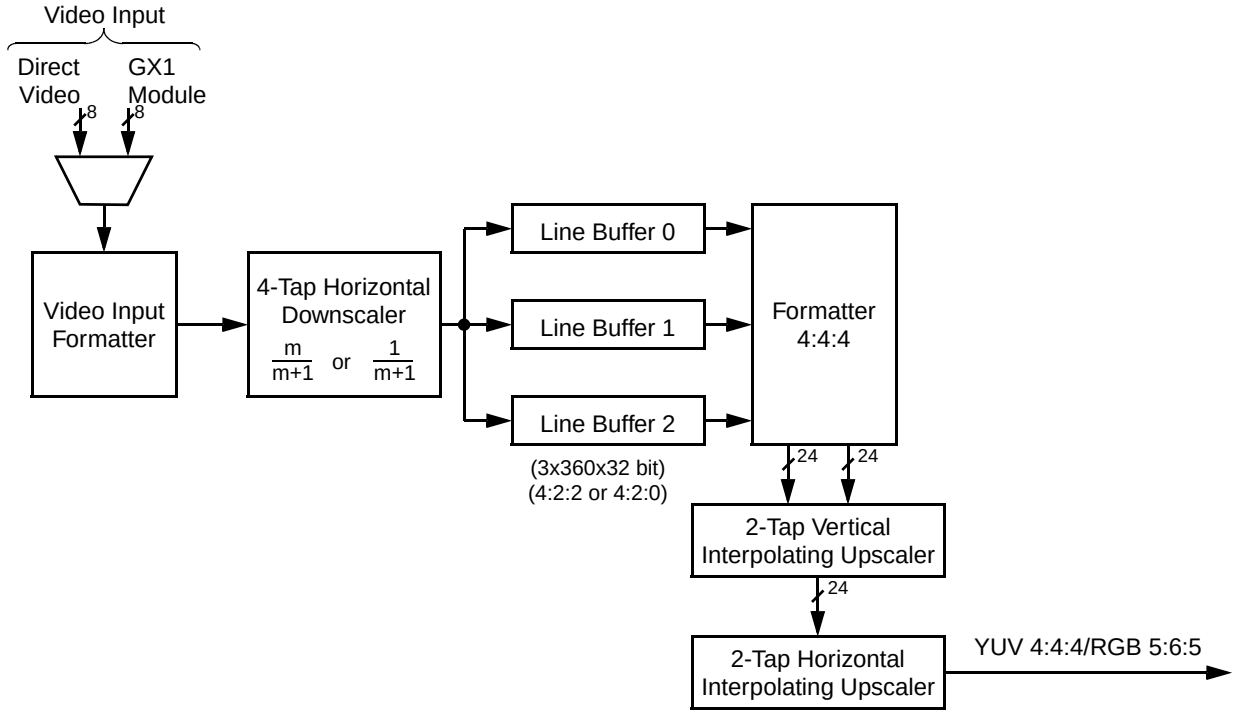


Figure 6-7. Video Block Diagram

Video Processor Module (Continued)

6.2.2.2 Horizontal Downscaler with 4-Tap Filtering

The Video Processor implements up to 8:1 horizontal downscaling with 4-tap filtering for horizontal interpolation. Filtering is performed on video data input to the Video Processor. This data is fed to the filter and then to the downscaler. There is a bypass path for both filtering and downscaling logic. If this bypass is enabled, video data is written directly into the line buffers. (See Figure 6-8.)

Filtering

There are four 4-bit coefficients which can have programmed values of 0 to 15. The filter coefficients can be programmed via the Video Downscaler Coefficient register (F4BAR0+Memory Offset 40h) to increase picture quality.

Horizontal Downscaler

The Video Processor supports horizontal downscaling. The downscaler can be implemented in the Video Processor to shrink the video window by a factor of up to 8:1, in 1-pixel increments. The downscaler factor (m) is programmed in the Video Downscaler Control register (F4BAR0+Memory Offset 3Ch[4:1]). If bit 0 of this register is set to 0, the downscaler logic is bypassed.

The horizontal downscaler supports downscaling of video data input format YUV 4:2:2 only.

The downscaler supports up to 29 downscaler factors. There are two types of factors:

- Type A is (1/m+1). One pixel is retained, and m pixels are dropped. This enables downscaling factors of 1/16, 1/15, 1/14, 1/13, 1/12, 1/11, 1/10, 1/9,1/8, 1/7, 1/6, 1/5, 1/4, 1/3, and 1/2.
- Type B is (m/m+1). m pixels are retained, and one pixel is dropped. This enables downscaling factors of 2/3, 3/4, 4/5, 5/6, 6/7, 7/8, 8/9, 9/10, 10/11, 11/12, 12/13, 13/14, 14/15, and 15/16.

Bit 6 of the Video Downscaler Control register (F4BAR0+Memory Offset 3Ch) selects the type of downscaling factor to be used.

Note: There is no vertical downscaling in the Video Processor.

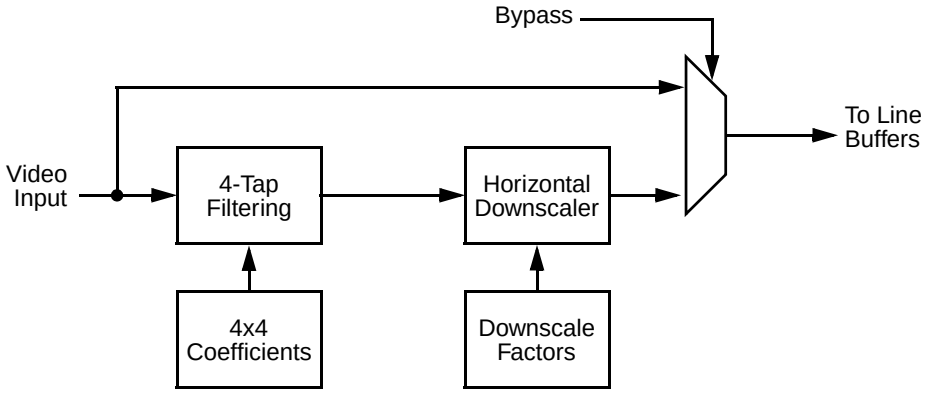


Figure 6-8. Horizontal Downscaler Block Diagram

Video Processor Module (Continued)

6.2.2.3 Line Buffers

After the data has been optionally horizontally downscaled the video data is stored in a 3-line buffer. Each line is 360 DWORDs, which means a line width of up to 720 pixels can be stored. This buffer supports two functions. First, the clock domain of the video data changes from the GX1 module's video clock to the GX1 module's graphics clock. This clock domain change is required because the video data and graphics data can only be mixed/blended in the same clock domain. The second function the line buffer performs is to provide the necessary look ahead and look behind data in the vertical direction for the vertical upscaler. There is no direct program control of the line buffer.

6.2.2.4 Formatter

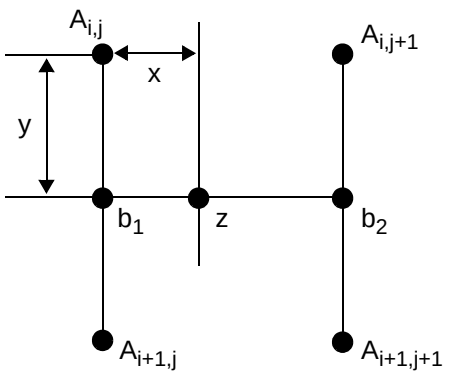
Video data in YUV 4:2:2 or YUV 4:2:0 format is converted to YUV 4:4:4 format. RGB data is not translated. There is no direct program control of the Formatter.

6.2.2.5 2-Tap Vertical and Horizontal Upscalers

After the video data has been buffered, the upscaling algorithm can be applied. The Video Processor employs a Digital Differential Analyzer-style (DDA) algorithm for both horizontal and vertical upscaling. The scaling parameters are programmed via the Video Upscale register (F4BAR0+Memory Offset 10h). The scalers support up to 8x factors for both horizontal and vertical scaling. The scaled video pixel stream is then passed through bi-linear interpolating filters (2-tap, 8-phase) to smooth the output video, significantly enhancing the quality of the displayed image.

The X and Y Upscaler uses the DDA and linear interpolating filter to calculate (via interpolation) the values of the pixels to be generated. The interpolation formula uses $A_{i,j}$, $A_{i,j+1}$, $A_{i+1,j}$, and $A_{i+1,j+1}$ values to calculate the value of intermediate points. The actual location of calculated points is determined by the DDA algorithm.

The location of each intermediate point is one of eight phases between the original pixels (see Figure 6-9).



Notes:

x and y are 0 - 7

$$b_1 = (A_{i,j})\frac{8-y}{8} + (A_{i+1,j})\frac{y}{8}$$

$$b_2 = (A_{i,j+1})\frac{8-y}{8} + (A_{i+1,j+1})\frac{y}{8}$$

$$z = (b_1)\frac{8-x}{8} + (b_2)\frac{x}{8}$$

Figure 6-9. Linear Interpolation Calculation

Video Processor Module (Continued)

6.2.3 Mixer/Blender Block

The Mixer/Blender block of the Video Processor module performs all the necessary functions to properly mix/blend the video data and the graphics data. These functions include Color Space Conversion (CSC), optional Gamma correction, color/chroma key, and the mixing/blending logic. See Figure 6-10 for block diagram of the Mixer/Blender block.

Video/Graphics mixing/blending must be performed in the RGB format. The YUV to RGB CSC (Section 6.2.3.1 on page 328) must be used on the video data if it is in YUV format. If Gamma Correction (see Section 6.2.3.2) on the video data is desired, it must be done in the color space of the input video data, which can be either YUV or RGB. If Gamma Correction on the graphics data is desired, it must be done in the color space of the input graphics data, which is RGB.

The video data can be in progressive or interlaced format, while the graphics data is always in the progressive format. The Mixer/Blender can mix/blend either format of video data with graphics data. F4BAR0+Memory Offset 4Ch[9] programs the mix/blend format. Considering the color space and the data format, the Mixer/Blender supports five types of mixing/blending. Some of the mixing/blending types have additional programming considerations to enable them to work optimally. The valid mixing/blending configurations are listed in see Table 6-1 on page 328 along with any additional programming requirements.

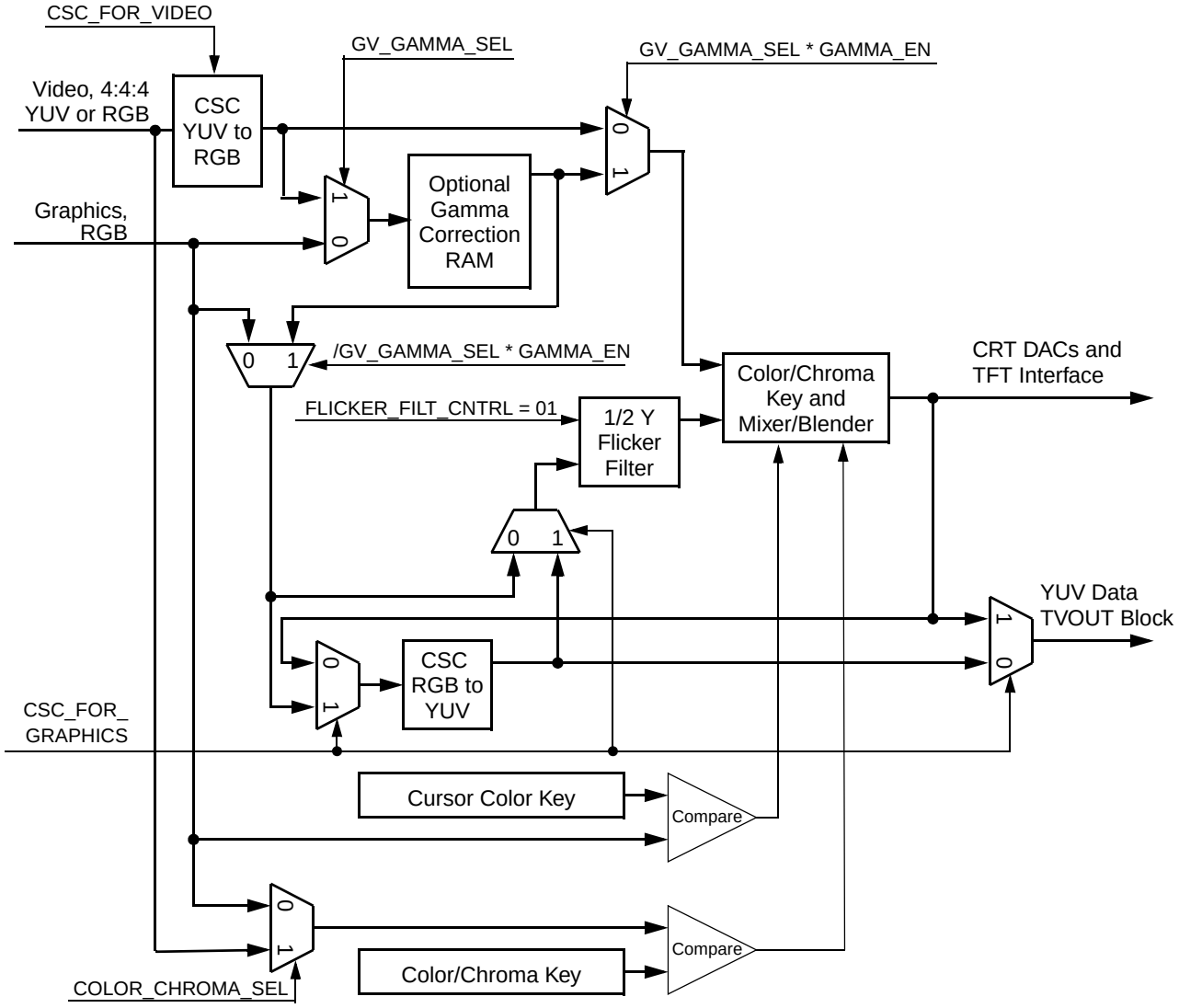


Figure 6-10. Mixer/Blender Block Diagram

Table 6-1. Valid Mixing/Blending Configurations

Mixing/Blending[1] (Bit)				Flicker Filter[2] (Bit)		Mode	Comment
13	11	10	9	30	29		
0	0	1	0	0	0	Input: YUV Progressive Video Mixing: RGB	• Produces highest quality RGB output (see Section 6.2.1.2 "Capture Video Mode", Weave subsection on page 322).
1	0	0	0	0	0	Input: RGB Progressive Video Mixing: RGB	• Produces highest quality RGB output (see Section 6.2.1.2 "Capture Video Mode", Weave subsection on page 322).
0	1	0	1	0	1	Input: YUV Interlaced Video Mixing: YUV	• Not supported.
0	1	0	0	0	0	Input: YUV Progressive Video Mixing: YUV	• Not supported.
0	0	1	0	0	0	Input: YUV Interlaced Video upscaled by 2 Mixing: RGB	• Typically Direct Video mode. • Must be vertically upscaled by a factor of 2 (see Section 6.2.2.5 "2-Tap Vertical and Horizontal Upscalers" on page 326).

1. F4BAR0+Memory Offset 4Ch[13, 11:9].
2. F4BAR0+Memory Offset 814h[30:29].

6.2.3.1 YUV to RGB CSC in Video Data Path

This CSC must be enabled if the video data is in the YUV color space. The CSC_FOR_VIDEO bit, F4BAR0+Memory Offset 4Ch[10], controls this CSC.

YUV video data is passed through this CSC to obtain 24-bit RGB data using the following CCIR-601-1 recommended formula:

- R = 1.1640625(Y – 16) + 1.59375(V – 128)
- G = 1.1640625(Y – 16) – 0.8125(V – 128) – 0.390625(U – 128)
- B = 1.1640625(Y – 16) + 2.015625(U – 128)

The CSC clamps inputs to prevent them from exceeding acceptable limits.

6.2.3.2 Gamma Correction

Either the video or graphics data can be routed through an integrated palette RAM for Gamma correction. There are three 256-byte RAMs, one for each color component value. Gamma correction supported in the YUV or RGB color space for the video data and RGB color space for the graphics data. Gamma correction is accomplished by treating each color component as an address into each RAM. The output of the RAM is the new color. A simple RGB Gamma correction example is to increase each color component by one. The address 00h in the RAMs would contain the data 01h. The address 01h would contain the data 02h and so on. This would have the effect of increasing each original Red, Green, and Blue value by one.

- G_V_GAMMA, F4BAR0+Memory Offset 04h[21] selects which data path (video or graphics) to send to the Gamma correction block. GAMMA_EN, F4BAR0+Memory Offset 28h[0] enables the Gamma correction function. To load the Gamma correction palette RAM, use F4BAR0+Memory Offset 1Ch and 20h.

6.2.3.3 Color/Chroma Key

A color/chroma key mechanism is used to support the Mixer/Blender logic. There are two keys: key1 is for the cursor and key2 is for graphics or video data. Key1, the cursor key, is always a color key. The cursor color key registers are located at, F4BAR0+Memory Offset 50h-5CF. How the cursor key mechanism works with the Mixer/Blender is explained in Section 6.2.3.4. COLOR_CHROMA_KEY (F4BAR0+Memory Offset 04h[20]) determines whether key2 is a color key or a chroma key. The Video Color Key Register (F4BAR0+Memory Offset 14h) stores the key. Color keying is used when video is overlaid on the graphics (GFX_INS_VIDEO, F4BAR0+Memory Offset 4Ch[8] = 0). Chroma keying is used when graphics is overlaid on the video (GFX_INS_VIDEO = 1). How the color/chroma key mechanism works with the Mixer/Blender is explained in Section 6.2.3.4.

Video Processor Module (Continued)

6.2.3.4 Color/Chroma Key and Mixer/Blender

The Mixer/Blender takes each pixel of the graphics and video data streams and mixes or blends them together. Mixing is simply choosing the graphics pixel or the video pixel. Blending takes a percentage of a graphics pixel (Alpha_value * Graphics_pixel_value) and percentage of the video pixel (1 - Alpha_Value * Video_pixel_value) and adds them together. The percentages of each add up to 100%. The actual formula is:

- Blended Pixel = (Alpha_value * Graphics_pixel_value) / 256 + ((256 – Alpha_value) * Video_pixel_value) / 256

Where: Alpha_value = 0 to 255

Mixing and blending are supported simultaneously for every rendered frame, however, each pixel can only be mixed or blended. The mix or blend question is decided by the pixel position, whether video is overlaid on the graphics or visa versa (GFX_INS_VIDEO, F4BAR0+Memory Offset 4Ch[8]), and several programmed “windows”. Figure 6-11 illustrates and example frame.

Graphics Window

The graphics window is defined in the GX1 module’s display controller and is always the full screen resolution.

Video Window

The video window tells the Mixer/Blender where the video window is and its size. If Direct Video mode is enabled (see Section 6.2.1.1 "Direct Video Mode" on page 320), the video window must be defined as the resolution of the video port data resolution (720x480 for NTSC, 720x576 for PAL). Vertical scaling is not allowed. Horizontal scaling is allowed. If the video source is from the GX1 module’s video frame buffer (which includes Capture Video mode, see Section 6.2.1.2 "Capture Video Mode" on page 321) then the video data can be scaled both horizontally and vertically. The video data size, scaled or unscaled, must equal the video window size. The Video X Position (horizontal) and Video Y Position (vertical) registers (F4BAR0+Memory Offset 08h and 0Ch) define the video window.

Cursor Window

The cursor window can be managed two ways: with the GX1 module’s hardware cursor or a software cursor. When using the hardware cursor, the displayed colors of the hardware cursor must be the cursor color keys (see Section 4.5.3 “Hardware Cursor” in the GX1 datasheet). When the software cursor is used, the cursor size and position are not defined using registers. The cursor size, position, and image are determined through the use of the cursor color key colors in the graphics frame buffer. When the cursor is described in this manner, the cursor can be of any size and shape.

Alpha Windows

Up to three alpha windows can be defined. They are used only for blending. They can be of any size up to the graphics window size and they may overlap. To support overlapping of the alpha windows they can be prioritized as to which one is on top (F4BAR0+Memory Offset 4Ch[20:16]). The alpha windows are programmed at F4BAR0+Memory Offset 60h-88h.

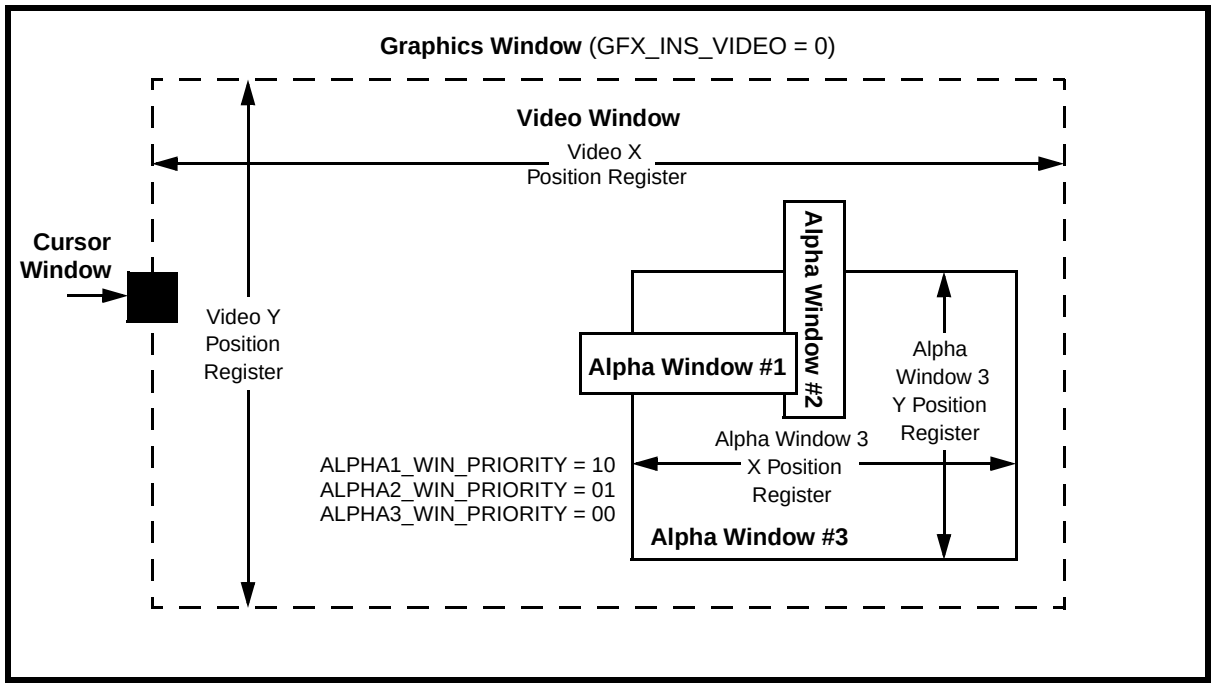


Figure 6-11. Graphics/Video Frame with Alpha Windows

Video Processor Module (Continued)

Mixing/Blending Operation

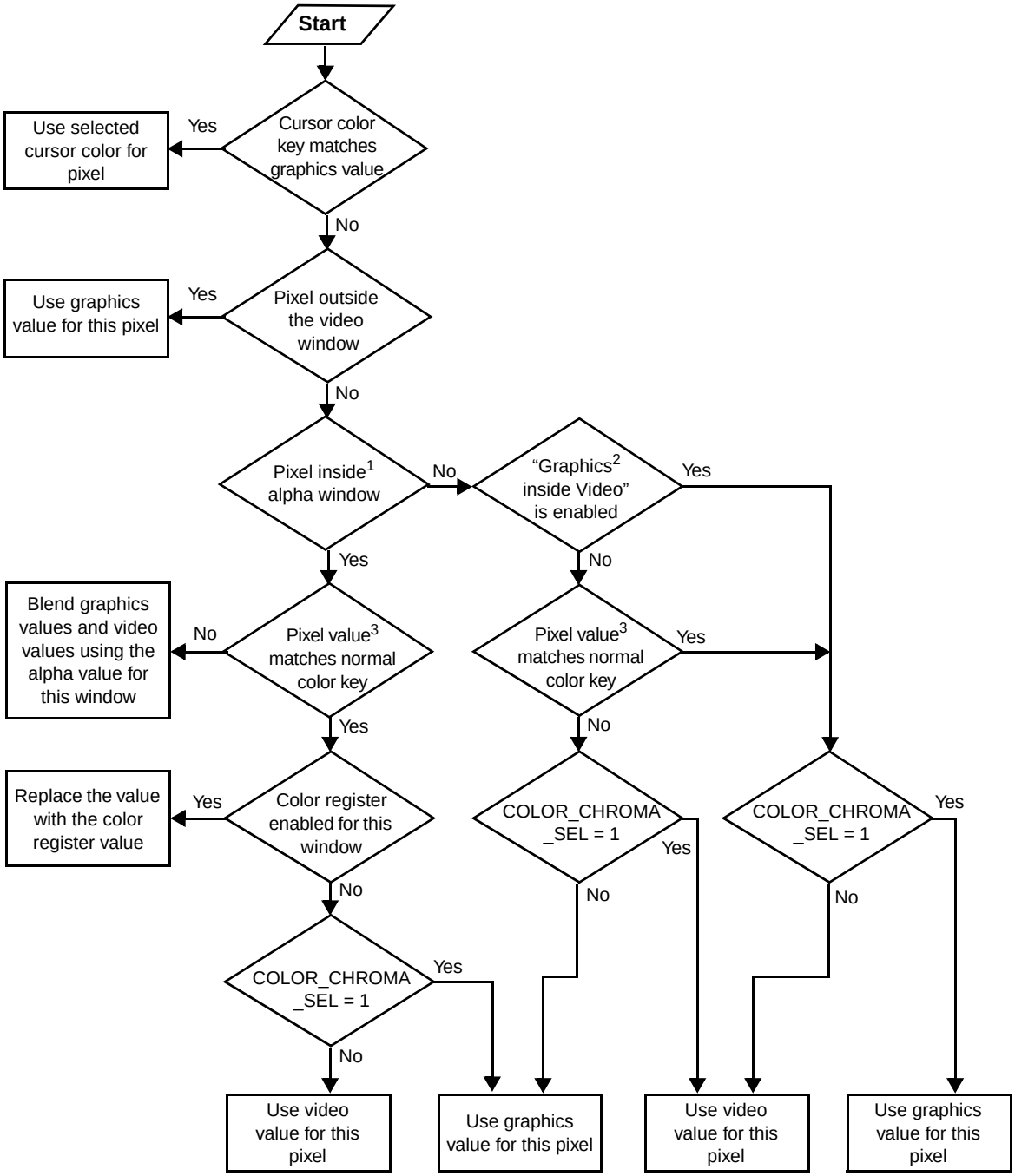
Table 6-2 on page 330 shows the truth table used to create the flow diagram, Figure 6-12 on page 331, that the Mixer/Blender logic uses to determine each pixels disposition.

Table 6-2. Truth Table for Alpha Blending

COLOR_ CHROMA_SEL[1]	Windows	Configuration[2]	Graphics Data Match Cursor Color Key	Graphics Data Match Normal Color Key	Video Data Match Normal Color Key	Mixer Output
x	x	x	Yes	x	x	Cursor Color
x	Not in Video Window	x	No	x	x	Graphics Data
Graphics Color Key (COLOR_ CHROMA_SEL = 0)	Not in an Alpha Window	GFX_INS_VIDEO = 0	No	Yes	x	Video Data
			No	No	x	Graphics Data
		GFX_INS_VIDEO = 1	No	x	x	Video Data
	Inside Alpha Window x	ALPHAx_COLOR_REG_EN = 1	No	Yes	x	Color from Color Register
		ALPHAx_COLOR_REG_EN = 0	No	Yes	x	Video Data
		x	No	No	x	Alpha-blended Data
Video Chroma Key (COLOR_ CHROMA_SEL = 1)	Not in an Alpha Window	GFX_INS_VIDEO = 0	No	x	Yes	Graphics Data
			No	x	No	Video Data
		GFX_INS_VIDEO = 1	No	x	x	Graphics Data
	Inside Alpha Window x	ALPHAx_COLOR_REG_EN = 1	No	x	Yes	Color from Color Register
		ALPHAx_COLOR_REG_EN = 0	No	x	Yes	Graphics Data
		x	No	x	No	Alpha-blended Data

1. COLOR_CHROMA_SEL: F4BAR0+Memory Offset 04h[20].
2. GFX_INS_VIDEO: F4BAR0+Memory Offset 4Ch[8].
 ALPHAx_COLOR_REG_EN: F4BAR0+Memory Offsets 68h[24], 78h[24], and 88h[24].

Video Processor Module (Continued)



Notes:

1) Alpha window should not be placed outside of the video window.

2) "Graphics inside Video" is enabled via bit GFX_INS_VIDEO in the Video De-interlacing and Alpha Control register (F4BAR0+Memory Offset 4Ch[8]).

3) The "Pixel Value" refers to either the Video value or the Graphics value, depending on the setting of bit COLOR_CHROMA_SEL in the Display Configuration register (F4BAR0+Memory Offset 04h[20]).

Figure 6-12. Color Key and Alpha Blending Logic

Video Processor Module (Continued)

6.2.4 VESA DDSC2B and DPMS Support

The Video Processor supports VESA, DDSC2B, and DPMS standards for enhanced monitor communications and power management support. This support is provided via signals DDC_SCL (muxed with IDE_DATA10) and DDC_SDA (muxed with IDE_DATA9). F4BAR0+Memory Offset 04h[24, 23, 22] controls the interface.

6.2.5 Integrated DACs

The Video Processor uses a Digital to Analog Converter (DAC) for CRT display.

To interface directly with the CRT display, the Video Processor incorporates triple 8-bit video DACs. The integrated DACs drive the RED, GREEN and BLUE inputs of the CRT. Each integrated DAC is an 8-bit current output type which can run at a clock rate of up to 135 MHz. The integrated DAC can generate voltage levels from 0 to 1.0V, when driving 75Ω double-terminated loads.

Differential and integral linearity errors, over full temperature and voltage ranges, are less than one LSB.

The peak white voltage (V_{FR} - full range output voltage), generated at the DAC, is defined according to the following formula:

$$V_{FR} = 3.35(V_{REF} / R_{SET}) * 75$$

where:

V_{REF} is the voltage at VREF (either internal bandgap reference, or externally connected voltage reference).

R_{SET} is the value of resistance between SETRES and AV_{SS} (typically 470Ω).

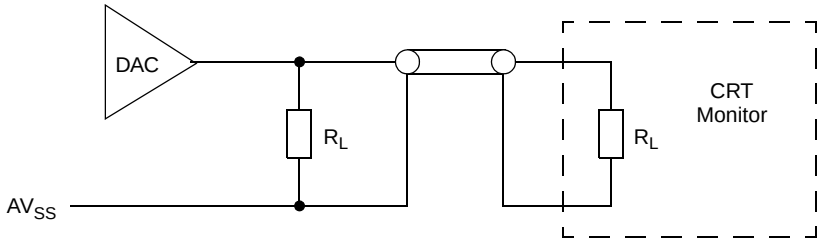


Figure 6-13. DAC Voltage Levels

Video Processor Module (Continued)

6.2.6 TFT Interface

The TFT interface can be programmed to one of two sets of balls: IDE balls or Parallel Port balls. PMR[23] of the General Configuration registers program where the TFT interface exists (see Table 3-2 on page 81).

Note: If the TFT interface is on the IDE balls, the maximum FPCLK supported is 40 MHz. If the TFT interface is on the Parallel Port balls the maximum FPCLK supported is 80 MHz.

Support for a TFT panel requires power sequencing and an 18-bit (6-bit RGB), digital output. The relevant digital output signals are available from the SC2200.

TFT output signals are:

- TFTD[5:0] for blue signals
- TFTD[11:6] for green signals
- TFTD[17:12] for red signals
- HSYNC and VSYNC - sync signals
- TFTDCK - data clock signal
- TFTDE - data enable signal
- FP_VDD_ON - power control signal

Power Sequence

Power sequence is used to control assertion of FP_VDD_ON and TFTD signals.

All bits related to power sequence configuration are located in the Display Configuration register (F4BAR0+Memory Offset 04h).

After enabling CRT_EN (bit 0), and FP_PWR_EN (bit 6), the state machine waits until the next VSYNC to switch on the FP_VDD_ON signal. The state machine then asserts the TFTD[17:0] signals after the delay programmed via PWR_SEQ_DLY (bits [19:17]) When FP_PWR_EN (bit 6) is set to 0, the reverse sequence happens for powering down the TFT.

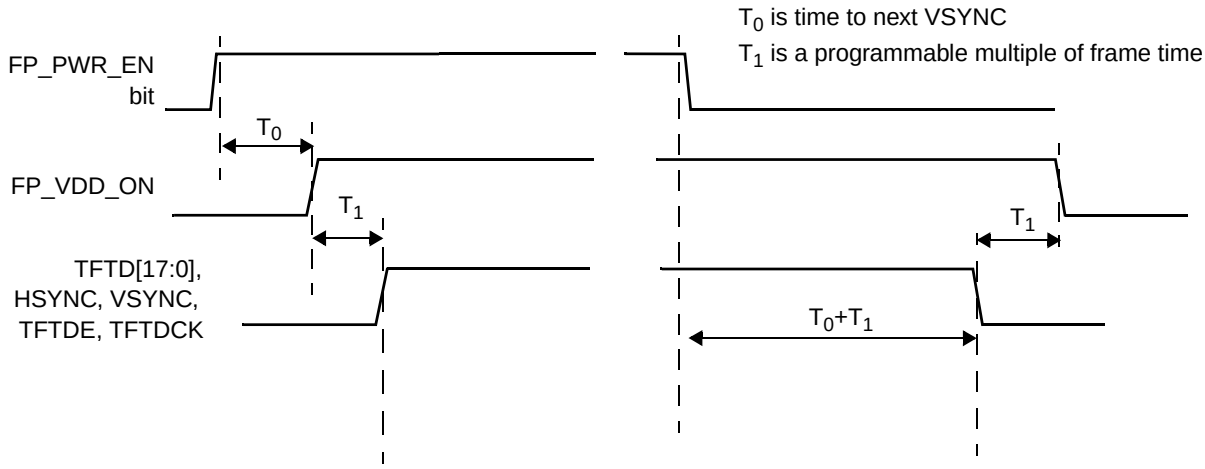


Figure 6-14. TFT Power Sequence

Video Processor Module (Continued)

6.2.7 Integrated PLL

The integrated (CRT) PLL can generate frequencies up to 135 MHz from a single 27 MHz source. The clock frequency is programmable using two registers. Figure 6-15 shows the block diagram of the Video Processor integrated PLL.

F_{REF} is 27 MHz, generated by an external crystal and an integrated oscillator. F_{OUT} is calculated from:

$$F_{OUT} = (m + 1) / (n+ 1) \times F_{REF}$$

The integrated PLL can generate any frequency by writing into the CRT-m and CRT-n bit fields (FBAR0+Memory Offset 2Ch). Additionally, 16 preprogrammed VGA frequencies can be selected via the PLL Clock Select register (F4BAR0+Memory Offset 2Ch[19:16]), if the crystal oscillator has a frequency of 27 MHz. This PLL can be powered down via the Miscellaneous register (F4BAR0+Memory Offset 28h[12]).

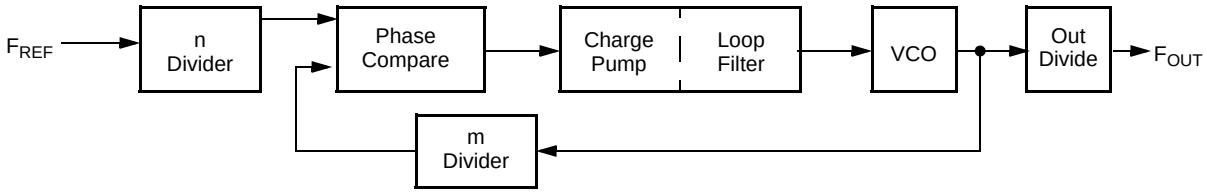


Figure 6-15. PLL Block Diagram

6.3 REGISTER DESCRIPTIONS

The register space for accessing and configuring the Video Processor is located in the Core Logic Chipset Register Space (F0-F5). The Chipset Register Space is accessed via the PCI interface using the PCI Type One Configuration Mechanism (see Section 5.3.1 "PCI Configuration Space and Access Methods" on page 184).

6.3.1 Register Summary

The tables in this subsection summarize the registers of the Video Processor. Included in the tables are the register's reset values and page references where the bit formats are found.

Table 6-3. F4: PCI Header Registers for Video Processor Support Summary

F4 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 6-6)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 338
02h-03h	16	RO	Device Identification Register	0504h	Page 338
04h-05h	16	R/W	PCI Command Register	0000h	Page 338
06h-07h	16	RO	PCI Status Register	0280h	Page 338
08h	8	RO	Device Revision ID Register	01h	Page 338
09h-0Bh	24	RO	PCI Class Code Register	030000h	Page 338
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 338
0Dh	8	RO	PCI Latency Timer Register	00h	Page 338
0Eh	8	RO	PCI Header Type Register	00h	Page 338
0Fh	8	RO	PCI BIST Register	00h	Page 338
10h-13h	32	R/W	Base Address Register 0 (F4BAR0). Sets the base address for the memory-mapped Video Configuration Registers within the Video Processor. Refer to Table 6-7 on page 340 for programming information regarding the register offsets accessed through this register.	00000000h	Page 338
14h-17h	32	R/W	Base Address Register 1 (F4BAR1). Reserved.	00000000h	Page 338
18h-1Bh	32	R/W	Base Address Register 2 (F4BAR2). Sets the base address for the memory-mapped VIP (Video Interface Port) Registers (summarized in Table 6-8 on page 353).	00000000h	Page 338
1Ch-2Bh	--	--	Reserved	00h	Page 338
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 338
2Eh-2Fh	16	RO	Subsystem ID	0504h	Page 338
30h-3Bh	--	--	Reserved	00h	Page 338
3Ch	8	R/W	Interrupt Line Register	00h	Page 338
3Dh	8	R/W	Interrupt Pin Register	03h	Page 339
3Eh-FFh	---	---	Reserved	00h	Page 339

Video Processor Module (Continued)

Table 6-4. F4BAR0: Video Processor Configuration Registers Summary

F4BAR0+ Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 6-7)
00h-03h	32	R/W	Video Configuration Register	00000000h	Page 340
04h-07h	32	R/W	Display Configuration Register	x0000000h	Page 341
08h-0Bh	32	R/W	Video X Position Register	00000000h	Page 342
0Ch-0Fh	32	R/W	Video Y Position Register	00000000h	Page 342
10h-13h	32	R/W	Video Upscaler Register	00000000h	Page 343
14h-17h	32	R/W	Video Color Key Register	00000000h	Page 343
18h-1Bh	32	R/W	Video Color Mask Register	00000000h	Page 343
1Ch-1Fh	32	R/W	Palette Address Register	xxxxxxxxh	Page 343
20h-23h	32	R/W	Palette Data Register	xxxxxxxxh	Page 344
24h-27h	32	RO	Reserved	---	Page 344
28h-2Bh	32	R/W	Miscellaneous Register	00001400h	Page 344
2Ch-2Fh	32	R/W	PLL2 Clock Select Register	00000000h	Page 344
30h-33h	32	---	Reserved	00000000h	Page 345
34h-37h	32	RO	Reserved	00000000h	Page 345
38h-3Bh	32	RO	Reserved	00000000h	Page 345
3Ch-3Fh	32	R/W	Video Downscaler Control Register	00000000h	Page 345
40h-43h	32	R/W	Video Downscaler Coefficient Register	00000000h	Page 345
44h-47h	32	R/W	CRC Signature Register	xxxxx100h	Page 345
48h-4Bh	32	RO	Device and Revision Identification	0000015xh	Page 346
4Ch-4Fh	32	R/W	Video De-Interlacing and Alpha Control Register	00060000h	Page 346
50h-53h	32	R/W	Cursor Color Key Register	00000000h	Page 347
54h-57h	32	R/W	Cursor Color Mask Register	00000000h	Page 347
58h-5Bh	32	R/W	Cursor Color Register 1	00000000h	Page 348
5Ch-5Fh	32	R/W	Cursor Color Register 2	00000000h	Page 348
60h-63h	32	R/W	Alpha Window 1 X Position Register	00000000h	Page 348
64h-67h	32	R/W	Alpha Window 1 Y Position Register	00000000h	Page 348
68h-6Bh	32	R/W	Alpha Window 1 Color Register	00000000h	Page 348
6Ch-6Fh	32	R/W	Alpha Window 1 Control Register	00000000h	Page 349
70h-73h	32	R/W	Alpha Window 2 X Position Register	00000000h	Page 349
74h-77h	32	R/W	Alpha Window 2 Y Position Register	00000000h	Page 349
78h-7Bh	32	R/W	Alpha Window 2 Color Register	00000000h	Page 349
7Ch-7Fh	32	R/W	Alpha Window 2 Control Register	00000000h	Page 350
80h-83h	32	R/W	Alpha Window 3 X Position Register	00000000h	Page 350
84h-87h	32	R/W	Alpha Window 3 Y Position Register	00000000h	Page 350
88h-8Bh	32	R/W	Alpha Window 3 Color Register	00000000h	Page 350
8Ch-8Fh	32	R/W	Alpha Window 3 Control Register	00000000h	Page 351
90h-93h	32	R/W	Video Request Register	001B0017h	Page 351
94h-97h	32	RO	Alpha Watch Register	00000000h	Page 351
98h-3FFh		---	Reserved	---	Page 351
400h-403h	32	R/W	Video Processor Display Mode Register	00000000h	Page 351
404h-407h	32	---	Reserved	00000000h	Page 352
408h-40Bh	32	R/W	Video Processor Test Mode Register	00000000h	Page 352
40Ch-41Fh	---	---	Reserved	00000000h	Page 352
420h-423h	32	R/W	GenLock Register	00000000h	Page 352
424h-427h	32	R/W	GenLock Delay Register	00000000h	Page 352
428h-43Bh	---	---	Reserved	---	Page 352
43Ch-43Fh	32	R/W	Continuous GenLock Time-out Register	1FFF1FFFh	Page 352

Video Processor Module (Continued)

Table 6-5. F4BAR2: VIP Support Registers Summary

F4BAR2+ Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 6-8)
00h-03h	32	R/W	Video Interface Port Configuration Register	00000000h	Page 353
04h-07h	32	R/W	Video Interface Control Register	00000000h	Page 353
08h-0Bh	32	R/W	Video Interface Status Register	xxxxxxxxh	Page 354
0Ch-0Fh	--	--	Reserved	00000000h	Page 355
10h-13h	32	RO	Video Current Line Register	xxxxxxxxh	Page 355
14h-17h	32	R/W	Video Line Target Register	00000000h	Page 355
18h-1Fh	---	---	Reserved	00000000h	Page 355
20h-23h	32	R/W	Video Data Odd Base Register	00000000h	Page 355
24h-27h	32	R/W	Video Data Even Base Register	00000000h	Page 355
28h-2Bh	32	R/W	Video Data Pitch Register	00000000h	Page 355
2Ch-3Fh	--	--	Reserved	00000000h	Page 355
40h-43h	32	R/W	VBI Data Odd Base Register	00000000h	Page 355
44h-47h	32	R/W	VBI Data Even Base Register	00000000h	Page 356
48h-4Bh	32	R/W	VBI Data Pitch Register	00000000h	Page 356
4Ch-1FFh	--	--	Reserved	00000000h	Page 356

Video Processor Module (Continued)

6.3.2 Video Processor Registers - Function 4

The register space designated as Function 4 (F4) is used to configure the PCI portion of support hardware for accessing the Video Processor support registers, including VIP (separate BAR). The bit formats for the PCI Header registers are given in Table 6-6.

Located in the PCI Header Registers of F4 are three Base Address Registers (F4BARx) used for pointing to the register spaces designated for Video Processor support. F4BAR0 is for Video Processor Configuration, F4BAR1 is reserved, and F4BAR2 is for VIP configuration.

Table 6-6. F4: PCI Header Registers for Video Processor Support Registers

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0504h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:2	**Reserved. (Read Only)**
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled to access memory offsets through F4BAR0, F4BAR1, and F4BAR2 (see F4 Index 10h, 14h, and 18h).
0	**Reserved. (Read Only)**
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 01h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 030000h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F4BAR0 (R/W)** — **Reset Value: 00000000h** **Video Processor Video Memory Address Space.** This register allows PCI access to the memory mapped Video Processor configuration registers. Bits [11:0] are read only (0000 0000 0000) indicating a 4 KB memory address range. See Table 6-7 on page 340 for bit formats and reset values of the registers accessed through this base address register.
31:12	**Video Processor Video Memory Base Address.**
11:0	**Address Range. (Read Only)**
Index 14h-17h	**Base Address Register 1 - F4BAR1 (R/W)** — **Reset Value: 00000000h** **Reserved**
Index 18h-1Bh	**Base Address Register 2 - F4BAR2 (R/W)** — **Reset Value: 00000000h** **VIP Address Space.** This register allows access to memory mapped VIP (Video Interface Port) related registers. Bits [11:0] are read only (0000 0000 0000), indicating a 4 KB I/O address range. Refer to Table 6-8 for the VIP register bit formats and reset values.
31:12	**VIP Base Address.**
11:0	**Address Range. (Read Only)**
Index 1Ch-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0504h**
Index 30h-3Bh	**Reserved** — **Reset Value: 00h**
Index 3Ch	**Interrupt Line Register (R/W)** — **Reset Value: 00h** This register identifies the system interrupt controllers to which the device's interrupt pin is connected. The value of this register is used by device drivers and has no direct meaning to this function.

Table 6-6. F4: PCI Header Registers for Video Processor Support Registers (Continued)

Bit	Description
Index 3Dh	**Interrupt Pin Register (R/W)** **Reset Value: 03h**
This register selects which interrupt pin the device uses. VIP uses INTC# after reset. INTA#, INTB# or INTD# can be selected by writing 1, 2 or 4, respectively.	
Index 3Eh-FFh	**Reserved** **Reset Value: 00h**

Video Processor Module (Continued)

6.3.2.1 Video Processor Support Registers - F4BAR0

F4 Index 10h, Base Address Register 0 (F4BAR0) sets the base address that allows PCI access to the Video Processor support registers, not including VIP. A separate base address register (F4BAR2) is used to access VIP support registers (see Section 6.3.2.2 on page 353).

Note: Reserved bits that are not defined as “must be set to 0 or 1" should be written with a value that is read from them.

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers

Bit	Description
Offset 00h-03h **Video Configuration Register (R/W)** **Reset Value: 00000000h** Configuration register for options of the motion video acceleration hardware.	
31:29	**Reserved.** Must be set to 0.
28	**EN_42X (Enable 4:2:x Format).** Allows format selection. 0: 4:2:2 format. 1: 4:2:0 format. **Note:** When input video stream is RGB (i.e., F4BAR0+Memory Offset 4Ch[13] = 1), this bit must be set to 0.
27	**BIT_8_LINE_SIZE.** When enabled, this bit increases line size from VID_LIN_SIZ (bits [15:8]) DWORDs by adding 256 DWORDs. 0: Disable. 1: Enable.
26:25	**Reserved.** Must be set to 0.
24:16	**INIT_RD_ADDR (Initial Buffer Read Address).** This field preloads the starting read address for the line buffers at the beginning of each display line. It is used for hardware clipping of the video window at the left edge of the active display. It represents the DWORD address of the source pixel which is to be displayed first. For an unclipped window, this value should be 0. For 4:2:0 format, set bits [17:16] to 00.
15:8	**VID_LIN_SIZ (Video Line Size).** Represents the number of DWORDs that make up the horizontal size of the source video data.
7	**YFILT_EN (Y Filter Enable).** Enables/disables the vertical filter. 0: Disable. Upscaling done by repeating pixels. 1: Enable. Upscaling done by interpolating pixels. **Note:** This bit is used with Y upscaling logic. Reset to 0 when not required.
6	**XFILT_EN (X Filter Enable).** Enables/disables the horizontal filter. 0: Disable. Upscaling done by repeating pixels. 1: Enable. Upscaling done by interpolating pixels. **Note:** This bit is used with X upscaling logic. Reset to 0 when not required.
5:4	**Reserved.**
3:2	**VID_FMT (Video Format).** Byte ordering of video data on the Video Input bus (VPD[7:0]). The interpretation of these bits depends on the settings of bit 13 (GV_SEL) in the Video De-Interlacing and Alpha Control register (F4BAR0+Memory Offset 4Ch) and bit 28 (EN_42X) of this register. If GV_SEL = 0 and EN_42X = 0: 00: Cb Y0 Cr Y1 10: Y0 Cb Y1 Cr 01: Y1 Cr Y0 Cb 11: Y0 Cr Y1 Cb If GV_SEL = 0 and EN_42X = 1: 00: Y0 Y1 Y2 Y3 10: Y1 Y0 Y3 Y2 01: Y3 Y2 Y1 Y0 11: Y1 Y2 Y3 Y0 If GV_SEL = 1 and EN_42X = 0: 00: P1L P1M P2L P2M 10: P1M P1L P2M P2L 01: P2M P2L P1M P1L 11: P1M P2L P2M P1L If GV_SEL = 1 and EN_42X = 1: Reserved **Note:** Both RGB 5:6:5 and YUV 4:2:2 contain two pixels in each 32-bit DWORD. YUV 4:2:0 contains a stream of Y data for each line, followed by U and V data for that same line.
1	**Reserved.**

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
0	**VID_EN (Video Enable).** Enables video acceleration hardware. 0: Disable (reset) video module. 1: Enable.
Offset 04h-07h	**Display Configuration Register (R/W)** **Reset Value: x0000000h** General configuration register for display control. This register is also used to determine how graphics and video data are to be combined in the display on the output device.
31	**DDC_SDA_IN (DDC Input Data). (Read Only)** Returns the value from the DDC_SDA signal (ball C20, muxed with IDE_DATA9) connected to pin 12 of the VGA connector.
30:28	**Reserved.**
27	**FP_ON_STATUS (Flat Panel On Status). (Read Only)** Shows whether power to the attached flat panel is on or off. This bit transitions at the end of the power-up or power-down sequence. 0: Power to the flat panel is off. 1: Power to the flat panel is on.
26	**DAC_VREF (CRT DAC Voltage Reference).** When set to 1, this bit enables use of an external voltage reference for CRT DAC. 0: Disable external VREF. Enable Internal VREF. 1: Use external VREF. Connect an external voltage reference to the VREF signal (ball P1).
25	**Reserved.** Must be set to 0.
24	**DDC_OE (DDC Output Enable).** Selects the direction of signal DDC_SDA (ball C20, muxed with IDE_DATA9). This bit indicates the direction of DDC data flow between the Video Processor and a CRT. 0: Input. 1: Output. DDC data is sent from the Video Processor to the CRT.
23	**DDC_SDA_OUT (DDC Output Data).** DDC data bit for output.
22	**DDC_SCL (DDC Serial Clock).** Provides the serial clock for the interface using the DDC_SCL signal (ball A20, muxed with IDE_DATA10).
21	**GV_GAMMA_SEL (Graphics or Video Gamma Source Data).** Selects whether the graphics or video data goes to the Gamma Correction RAM. GAMMA_EN (F4BAR0+Memory Offset 28h[0]) must be enabled for the selected data source to pass through the Gamma Correction RAM. 0: Graphics data to Gamma Correction RAM. 1: Video data to Gamma Correction RAM. **Note:** Gamma Correction is always in the RGB domain for graphics data. Gamma Correction can be in the YUV or RGB domain for video data.
20	**COLOR_CHROMA_SEL (Color or Chroma Key Select).** Selects whether the graphics is used for color keying or the video data stream is used for chroma keying. 0: Graphics data is compared to the color key. 1: Video data is compared to the chroma key.
19:17	**PWR_SEQ_DLY (Power Sequence Delay).** Selects the number of frame periods that transpire between successive transitions of the power sequence control lines.
16:14	**CRT_SYNC_SKW (CRT Sync Skew).** Represents the number of pixel clocks to skew the horizontal and vertical sync that are sent to the CRT. This field should be programmed to 100 at the baseline. Via this register, the sync can be moved forward (later) or backward (earlier) relative to the pixel data. This register can be used to compensate for possible delay of pixel data being processed via the Video Processor. 000: Sync moved 4 clocks backward — 100: Baseline, sync not moved 001: Sync moved 3 clocks backward — 101: Sync moved 1 clock forward 010: Sync moved 2 clocks backward — 110: Sync moved 2 clocks forward 011: Sync moved 1 clock backward — 111: Sync moved 3 clocks forward
13:10	**Reserved.**
9	**CRT_VSYNC_POL (CRT Vertical Synchronization Polarity).** Selects CRT vertical sync polarity. 0: CRT vertical sync is normally low, and is set high during the sync interval. 1: CRT vertical sync is normally high, and is set low during the sync interval.

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
8	**CRT_HSYNC_POL (CRT Horizontal Synchronization Polarity).** Selects CRT horizontal sync polarity. 0: CRT horizontal sync is normally low, and is set high during sync interval. 1: CRT horizontal sync is normally high, and is set low during sync interval.
7	**FP_DATA_EN (Flat Panel Output Enable).** Controls the data, data-enable, clock and sync output signals. 0: Flat panel data outputs are forced to zero depending on the value of bit 3 (DAC_BL_EN). Bit 6 (FP_PWR_EN**)** is ignored. 1: Flat panel outputs are forced to zero until power-up, and later, data outputs are subject to the value of bit 3 (DAC_BL_EN).
6	**FP_PWR_EN (Flat Panel Power Enable).** Changing this bit initiates a flat panel power-up or power-down. 0-to-1: Power-up flat panel. 1-to-0: Power-down flat panel.
5:4	**Reserved.**
3	**DAC_BL_EN (DAC Blank Enable).** Controls blanking of the CRT DACs. 0: DACs are constantly blanked. 1: DACs are blanked normally (i.e., during horizontal and vertical blank).
2	**VSYNC_EN (Vertical Sync Enable).** Enables/disables display vertical sync (used for VESA DPMS support). 0: Disable. 1: Enable.
1	**HSYNC_EN (Horizontal Sync Enable).** Enables/disables display horizontal sync (used for VESA DPMS support). 0: Disable. 1: Enable.
0	**CRT_EN (CRT Enable).** Enables the CRT control logic. This bit is also used to reset the CRT control logic. 0: Reset CRT control logic. 1: Enable CRT control logic.

Offset 08h-0Bh **Video X Position Register (R/W)** **Reset Value: 00000000h**

Provides the window X position. This register is programmed relative to CRT horizontal sync input (not physical screen position).

Note: H_TOTAL and H_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8330h[26:19] and 8338h[10:3], respectively). The value of (H_TOTAL – H_SYNC_END) is sometimes referred to as "horizontal back porch". For more information, see the *GX1 Processor Series Datasheet*.

31:28	**Reserved.**
27:16	**VID_X_END (Video X End Position).** Represents the horizontal end position of the video window (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 13.
15:12	**Reserved.**
11:0	**VID_X_START (Video X Start Position).** Represents the horizontal start position of the video window. This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 14.

Offset 0Ch-0Fh **Video Y Position Register (R/W)** **Reset Value: 00000000h**

Provides the window Y position. This register is programmed relative to CRT vertical sync input (not physical screen position).

Note: V_TOTAL and V_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8340h[26:16] and 8348h[26:16], respectively). The value of (V_TOTAL – V_SYNC_END) is sometimes referred to as "vertical back porch". For more information, see the *GX1 Processor Series Datasheet*.

31:27	**Reserved.**
26:16	**VID_Y_END (Video Y End Position).** Represents the vertical end position of the video window (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 2.
15:11	**Reserved**
10:0	**VID_Y_START (Video Y Start Position).** Represents the vertical start position of the video window. This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 1.

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
Offset 10h-13h	**Video Upscale Register (R/W)** — **Reset Value: 00000000h**
Provides horizontal and vertical upscale factors of the window.	
31:30	**Reserved.**
29:16	**VID_Y_SCL (Video Y Scale Factor).** Represents the vertical upscale factor of the video window according to the following formula: VID_Y_SCL = 8192 * (Ys - 1) / (Yd - 1) where: Ys = Video source vertical size in pixels Yd = Video destination vertical size in pixels **Note:** Upscale factor must be used. Yd is equal or bigger than Ys. If no scaling is intended, set to 2000h. The actual scale factor used is VID_Y_SCL/8192, but the formula above fits a given source number of lines into a destination window size. **Note:** When progressive mixing/blending is programmed (F4BAR0+Memory Offset 4Ch[9] = 0) and the video data is interlaced, this register should be programmed to 1000h to double the vertical lines,
15:14	**Reserved.**
13:0	**VID_X_SCL (Video X Scale Factor).** Represents horizontal upscale factor of the video window according to the following formula: VID_X_SCL = 8192 * (Xs - 1) / (Xd - 1) where: Xs = Video source horizontal size in pixels Xd = Video destination vertical size in pixels **Note:** Upscale factor must be used. Xd is equal or bigger than Xs. If no scaling is intended, set to 2000h. The actual scale factor used is VID_X_SCL/8192, but the formula above fits a given source number of pixels into a destination window size.
Offset 14h-17h	**Video Color Key Register (R/W)** — **Reset Value: 00000000h**
Provides the video color key. The color key can be used to allow irregular shaped overlays of graphics onto video, or video onto graphics, within a scaled video window.	
31:24	**Reserved.**
23:0	**VID_CLR_KEY (Video Color Key).** The video color key is a 24-bit RGB or YUV value. • If the COLOR_CHROMA_SEL bit (F4BAR0+Memory Offset 04h[20]) = 0: — The video pixel is selected within the target window if the corresponding graphics pixel matches the color key. The color key in an RGB value. • If the COLOR_CHROMA_SEL bit (F4BAR0+Memory Offset 04h[20]) = 1: — The video pixel is selected within the target window only if it (the video pixel) does not match the color key. The color key is usually an RGB value. However, if both the CSC_for VIDEO and GV_SEL bits (F4BAR0+Memory Offset 4Ch bits 10 and 13, respectively) are programmed to 0, the color key is a YUV value (i.e., video is not converted to RGB). The graphics or video data being compared can be masked prior to the compare via the Video Color Mask register (described in F4BAR0+Memory Offset 18h).
Offset 18h-1Bh	**Video Color Mask Register (R/W)** — **Reset Value: 00000000h**
Provides the video color mask. This value is used to mask bits of the graphics or video stream being compared to the video color key (described in F4BAR0+Memory Offset 14h). It can be used to allow a range of values to serve as the color key.	
31:24	**Reserved.**
23:0	**VID_CLR_MASK (Video Color Mask).** This mask is a 24-bit value. Zeros in the mask cause the corresponding bits in the graphics or video stream to be ignored.
Offset 1Ch-1Fh	**Palette (Gamma Correction RAM) Address Register (R/W)** — **Reset Value: xxxxxxxxh**
31:8	**Reserved.**
7:0	**PAL_ADDR (Palette Address).** Specifies the address to be used for the next access to the Palette Data register (F4BAR0+Memory Offset 20h[31:8]). Each access to the data register automatically increments the Palette Address register. If non-sequential access is made to the palette, the address register must be loaded between each non-sequential data block.

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
Offset 20h-23h	**Palette (Gamma Correction RAM) Data Register (R/W)** — **Reset Value: xxxxxxxxh**
	Provides the video palette data. The data can be read or written to the Gamma Correction RAM (palette) via this register. Prior to accessing this register, an appropriate address should be loaded to the Palette Address register (F4BAR0+Memory Offset 1Ch[7:0]). Subsequent accesses to the Palette Data register cause the internal address counter to be incremented for the next cycle.
31:8	**PAL_DATA (Palette Data).** Contains the read or write data for a Gamma Correction RAM (palette). **Note:** When a read or write to the Gamma Correction RAM occurs, the previous output value is held for one additional DOTCLK period. This effect should go unnoticed during normal operation.
7:0	**Reserved.**
Offset 24h-27h	**Reserved**
Offset 28h-2Bh	**Miscellaneous Register (R/W)** — **Reset Value: 00001400h**
	Configuration and control register for miscellaneous characteristics of the Video Processor.
31:13	**Reserved.**
12	**PLL2_PWR_EN (PLL2 Power-Down Enable).** 0: Power-down. 1: Normal.
11	**A_PWR_DN (Analog Power-Down).** Enables power-down of the PLL2 and the bandgap circuit that generates VREF. 0: Normal. 1: Power-down. **Note:** If A_PWR_DN is set to 1 without also setting DAC_PWR_DN (bit 10) to 1, an unexpected increase in power consumption may result.
10	**DAC_PWR_DN (DAC Power-Down).** Powers down the internal CRT DAC. 0: Normal. 1: Power-down.
9:1	**Reserved.**
0	**GAMMA_EN (Gamma Correction RAM Enable).** Allows video or graphics (selected by F4BAR0+Memory Offset 04h[21]) to go to the Gamma Correction RAM. 0: Enable. 1: Disable.
Offset 2Ch-2Fh	**PLL2 Clock Select Register (R/W)** — **Reset Value: 00000000h**
	Determines the characteristics of the integrated PLL2.
31:23	**Reserved.** Must be set to 0.
22:21	**CLK_DIV_SEL (Clock Divider Select).** 00: No division. 01: Divide by 2. 10: Divide by 4. 11: Divide by 8. Divides the clock generated by the PLL2, using the programmed m (bits [14:8]) and n (bits [3:0]) values.
20	**SEL_REG_CAL.** Selects specific or previously-calculated values. 0: Values previously calculated from the CLK_SEL bits (bits [19:16]). 1: Values according to the m (bits [14:8]), n (bits [3:0]), and CLK_DIV_SEL (bits [22:21]) fields.
19:16	**CLK_SEL (Clock Select).** Selects frequency (in MHz) of the display clock. 0000: 25.175 0100: 50 1000: 65 1100: 108 0001: 31.5 0101: 49.5 1001: 75 1101: 135 0010: 36 0110: 56.25 1010: 78.5 1110: 27 0011: 40 0111: 44.9 1011: 94.5 1111: 24.923052
15	**LFTC (Loop Filter Time Constant).** This bit should be set when m (bits [14:8]) value is higher than 30.

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
14:8	**m (Defines m PLL2 Value).** Relevant when SEL_REG_CAL (bit 20) = 1. The following formula is used for calculating the frequency using m and n values: Fvco = OSCCLK * Km/Kn Km = m + 1 Kn = n + 1 OSCCLK = 27 MHz
7:4	**Reserved**.
3:0	**n (Defines n PLL2 Value).** Relevant when SEL_REG_CAL (bit 20) = 1. The following formula is used for calculating the frequency using m and n values: Fvco = OSCCLK * Km/Kn Km = m + 1 Kn = n + 1 OSCCL = 27 MHz
Offset 30h-33h	**Reserved** **Reset Value: 00000000h**
Offset 34h-37h	**Reserved** **Reset Value: 00000000h**
Offset 38h-3Bh	**Reserved** **Reset Value: 00000000h**
Offset 3Ch-3Fh	**Video Downscaler Control Register (R/W)** **Reset Value: 00000000h**
Controls the characteristics of the integrated video downscaler.	
31:7	**Reserved.**
6	**DTS (Downscale Type Select).** 0: Type A (Downscale formula is 1/m+1, m pixels are dropped, 1 pixel is kept). 1: Type B (Downscale formula is m/m+1, m pixels are kept, 1 pixel is dropped).
5	**Reserved.**
4:1	**DFS (Downscale Factor Select).** Determines the downscale factor to be programmed into these bits, where m is used to derive the desired downscale factor depending on bit 6 (DTS).
0	**DCF (Downscaler and Filtering).** Enables/disables downscaler and filtering logic. 0: Disable. 1: Enable. **Note:** No downscaling support for RGB 5:6:5 and YUV 4:2:0 video formats.
Offset 40h-43h	**Video Downscaler Coefficient Register (R/W)** **Reset Value: 00000000h**
Indicates filter coefficients. The filters can be programmed independently to increase video quality when the downscaler is implemented. Valid values for each filter coefficient are 0-15. The sum of coefficients must be 16. FLT_CO_4 is used with the earliest pixels and FLT_CO_1 is used with the latest. Only luminance values of pixels are filtered.	
31:28	**Reserved.**
27:24	**FLT_CO_4 (Filter Coefficient 4).** For the tap-4 filter.
23:20	**Reserved.**
19:16	**FLT_CO_3 (Filter Coefficient 3).** For the tap-3 filter.
15:12	**Reserved.**
11:8	**FLT_CO_2 (Filter Coefficient 2).** For the tap-2 filter.
7:4	**Reserved.**
3:0	**FLT_CO_1 (Filter Coefficient 1).** For the tap-1 filter.
Offset 44h-47h	**CRC Signature Register (R/W)** **Reset Value: xxxxx100h**
Signature values stored in this register can be read by the host. This register is used for test purposes.	
31:8	**SIG_VALUE (Signature Value). (Read Only)** A 24-bit signature value is stored in this bit field and can be read at any time. The signature is produced from the RGB data output of the mixer. This bit field is used for test purpose only. See SIGN_EN (bit 0) description for more information.
7:3	**Reserved.**

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
2	**SIGN_FREE (Signature Free Run).** 0: Disable. (Default) If this bit was previously set to 1, the signature process stops at the end of the current frame (i.e., at the next falling edge of VSYNC). 1: Enable. If SIGN_EN (bit 0) = 1, the signature register captures data continuously across multiple frames.
1	**Reserved.**
0	**SIGN_EN (Signature Enable).** 0: Disable. (Default) The SIG_VALUE (bits [31:8]) is reset to 000001h and held (no capture). 1: Enable. The next falling edge of VSYNC is counted as the start of the frame to be used for CRC checking with each pixel clock beginning with the next VSYNC. If SIGN_FREE (bit 2) = 1, the signature register captures the pixel data signature continuously across multiple frames. If SIGN_FREE (bit 2) = 0, a signature is captured for one frame at a time, starting from the next falling VSYNC. After a signature capture, the SIG_VALUE can be read to determine the CRC check status. SIGN_EN can then be reset to initialize the SIG_VALUE as an essential preparation for the next round of CRC check.
Offset 48h-4Bh	**Device and Revision Identification (RO)** — **Reset Value: 0000xxxxh**
31:16	**Reserved.**
15:8	**REV_ID (Revision ID).** See device errata for value.
7:0	**DEV_ID (Device ID).** See device errata for value.
Offset 4Ch-4Fh	**Video De-Interlacing and Alpha Control Register (R/W)** — **Reset Value: 00060000h**
31:22	**Reserved.**
21:20	**ALPHA3_WIN_PRIORITY (Alpha Window 3 Priority).** Determines the priority of Alpha Window 3. A higher number indicates a higher priority. Priority is used to determine display order for overlapping alpha windows. 00: Lowest priority (default). 01: Medium priority. 10: Highest priority. 11: Illegal. **Note:** Priority of enabled alpha windows must be different.
19:18	**ALPHA2_WIN_PRIORITY (Alpha Window 2 Priority).** Determines the priority of Alpha Window 2. A higher number indicates a higher priority. Priority is used to determine display order for overlapping alpha windows. 00: Lowest priority (default). 01: Medium priority. 10: Highest priority. 11: Illegal. **Note:** Priority of enabled alpha windows must be different.
17:16	**ALPHA1_WIN_PRIORITY (Alpha Window 1 Priority).** Determines the priority of Alpha Window 1. A higher number indicates a higher priority. Priority is used to determine display order for overlapping alpha windows. 00: Lowest priority (default). 01: Medium priority. 10: Highest priority. 11: Illegal. **Note:** Priority of enabled alpha windows must be different.
15:14	**Reserved.**
13	**GV_SEL (GV Select).** Selects input video format. 0: YUV format. 1: RGB format. **Note:** Mixing and blending configurations are created using bits [13, 11:9] of this register. See Table 6-1 "Valid Mixing/Blending Configurations" on page 328. If this bit is set to 1, EN_42X (F4BAR0+Memory Offset 00h[28]) must be programmed to 0.

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
12	**VID_LIN_INV (Video Line Invert).** When this bit is set, it allows the video window to be positioned at odd offsets with respect to the first line. The values below are recommended if VID_Y_START (F4BAR0+Memory Offset 0Ch[10:0]) is an odd (set to 1) or even (set to 0) number of lines from the start of the active display. 0: Even. 1: Odd.
11	**Reserved.** Set to 0.
10	**CSC_FOR_VIDEO (Color Space Converter for Video).** Determines whether or not the video stream from the video module is passed through the CSC. 0: Disable. The video stream is sent "as is" to the video Mixer/Blender. 1: Enable. The video stream is passed through the CSC (for YUV to RGB conversion). **Note:** Mixing and blending configurations are created using bits [13, 11:9] of this register. See Table 6-1 "Valid Mixing/Blending Configurations" on page 328.
9	**VIDEO_BLEND_MODE (Video Blending Mode).** Allows selection of the type of video (i.e., interlaced or progressive) used for blending. 0: Progressive video used for blending. 1: Interlaced video used for blending. **Note:** Mixing and blending configurations are created using bits [13, 11:9] of this register. See Table 6-1 "Valid Mixing/Blending Configurations" on page 328.
8	**GFX_INS_VIDEO (Graphics Inside Video).** This bit works in conjunction with bit COLOR_CHROMA_SEL (F4BAR0+Memory Offset 4Ch[20]). COLOR_CHROMA_SEL selects whether the graphics is used for color keying or the video data stream is used for chroma keying. If COLOR_CHROMA_SEL = 0, graphics data is compared to the color key. If COLOR_CHROMA_SEL = 1, video data is compared to the chroma key. 0: Outside the alpha windows, graphics or video is displayed depending on the result of the color key comparison. 1: Outside the alpha windows, only video is displayed (if COLOR_CHROMA_SEL = 0) or only graphics is displayed (if COLOR_CHROMA_SEL = 1) color key comparison is not performed outside the alpha windows.
7	**VID_WIN_PUSH_EN (Video Window Push Enable).** Video window repositioning at an offset of 1 line below the programmed value. Facilitates line rate matching in both fields. 0: Disable. (Default) 1: Enable.
6	**TOP_LINE_IN_ODD (Top Line in Odd Field).** Allows selection of what field the top line is in. 0: Top line is in even field. (Default) 1: Top line is in odd field.
5	**Reserved.**
4	**INSERT_EN (Insert Enable).** When this bit is set, the odd frame is shifted with respect to the even frame. 0: No shifting occurs. 1: The odd frame is shifted according to the offset specified in bits [2:0].
3	**Reserved.**
2:0	**OFFSET (Vertical Scaler Offset).** For a non-interlaced video stream and when bob de-interlacing is used, program a value of 100 (i.e., shift one line); otherwise, leave at 000.

Offset 50h-53h	**Cursor Color Key Register (R/W)**	**Reset Value: 00000000h**
31:29	**Reserved.**	
28:24	**COLOR_REG_OFFSET (Cursor Color Register Offset).** This field indicates a bit in the incoming graphics stream. It is used to indicate which of the two possible cursor color registers should be used for color key matches for the bits in the graphics stream.	
23:0	**CUR_COLOR_KEY (Cursor Color Key).** Specifies the 24-bit RGB value of the cursor color key. The incoming graphics stream is compared with this value. If a match is detected, the pixel is replaced by a 24-bit value from one of the Cursor Color registers.	

Offset 54h-57h	**Cursor Color Mask Register (R/W)**	**Reset Value: 00000000h**
31:24	**Reserved.**	
23:0	**CUR_COLOR_MASK (Cursor Color Mask).** This mask is a 24-bit value. Zeroes in the mask cause the corresponding bits in the incoming graphics stream to be ignored.	

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
Offset 58h-5Bh	**Cursor Color Register 1 (R/W)** — **Reset Value: 00000000h**
31:24	**Reserved.**
23:0	**CUR_COLOR_REG1 (Cursor Color Register 1).** Specifies a 24-bit cursor color value. This is an RGB value (for RGB blending) or a YUV value (for YUV blending). In interlaced YUV blending mode, Y/2 value should be used. This is one of two possible cursor color values. The COLOR_REG_OFFSET bits (F4BAR0+Memory Offset 50h[28:24]) determine a bit of the graphics data that if even, selects this color to be used.
Offset 5Ch-5Fh	**Cursor Color Register 2 (R/W)** — **Reset Value: 00000000h**
31:24	**Reserved**
23:0	**CUR_COLOR_REG2 (Cursor Color Register 2).** Specifies a 24-bit cursor color value. This is an RGB value (for RGB blending) or a YUV value (for YUV blending). In interlaced YUV blending mode, Y/2 value should be used. This is one of two possible cursor color values. The COLOR_REG_OFFSET bits (F4BAR0+Memory Offset 50h[28:24]) determine a bit of the graphics data that if even, selects this color to be used.
Offset 60h-63h	**Alpha Window 1 X Position Register (R/W)** — **Reset Value: 00000000h**
Note:	H_TOTAL and H_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8330h[26:19] and 8338h[10:3], respectively). The value of (H_TOTAL – H_SYNC_END) is sometimes referred to as "horizontal back porch". For more information, see the *GX1 Processor Series Datasheet*. Desired screen position should not be outside a video window (F4BAR0+Memory Offset 08h and 0Ch).
31:27	**Reserved.**
26:16	**ALPHA1_X_END (Alpha Window 1 Horizontal End).** Determines the horizontal end position of Alpha Window 1 (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 1.
15:11	**Reserved.**
10:0	**ALPHA1_X_START (Alpha Window 1 Horizontal Start).** Determines the horizontal start position of Alpha Window 1. This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 2.
Offset 64h-67h	**Alpha Window 1 Y Position Register (R/W)** — **Reset Value: 00000000h**
Note:	V_TOTAL and V_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8340h[26:16] and 8348h[26:16], respectively). The value of (V_TOTAL – V_SYNC_END) is sometimes referred to as "vertical back porch". For more information, see the *GX1 Processor Series Datasheet*. Desired screen position should not be outside a video window (F4BAR0+Memory Offset 08h and 0Ch).
31:27	**Reserved.**
26:16	**ALPHA1_Y_END (Alpha Window 1 Vertical End).** Determines the vertical end position of Alpha Window 1 (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 2.
15:11	**Reserved.**
10:0	**ALPHA1_Y_START (Alpha Window 1 Vertical Start).** Determines the vertical start position of Alpha Window 1. This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 1.
Offset 68h-6Bh	**Alpha Window 1 Color Register (R/W)** — **Reset Value: 00000000h**
31:25	**Reserved.**
24	**ALPHA1_COLOR_REG_EN (Alpha Window 1 Color Register Enable).** Enable bit for the color key matching in Alpha Window 1. 1: Enable. If this bit is enabled and the alpha window is enabled, then where there is a color key match. The color value (in bits [23:0], ALPHA1_COLOR_REG) is displayed. 0: Disable. Where there is a color key match, no blending is performed.
23:0	**ALPHA1_COLOR_REG (Alpha Window 1 Color Register).** Specifies the color to be displayed inside Alpha Window 1 when there is a color key match in the alpha window. This is an RGB value (for RGB blending) or a YUV value (for YUV blending). In interlaced YUV blending mode, Y/2 value should be used. This color is only displayed if the alpha window is enabled and bit 24 (ALPHA1_COLOR_REG_EN) is enabled.

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
Offset 6Ch-6Fh	**Alpha Window 1 Control Register (R/W)** — **Reset Value: 00000000h**
31:18	**Reserved.**
17	**LOAD_ALPHA (Load Alpha Value). (Write Only)** When set to 1, this bit causes the Video Processor to load the alpha value (in bits [7:0], ALPHA_VAL) at the start of the next frame.
16	**ALPHA1_WIN_EN (Alpha Window 1 Enable).** Enable bit for Alpha Window 1. 1: Enable Alpha Window 1. 0: Disable Alpha Window 1. **Note:** Valid only if video window is enabled (F4BAR0+Memory Offset 00h[0] = 1).
15:8	**ALPHA1_INC (Alpha Window 1 Increment).** Specifies the alpha value increment/decrement. This is a signed 8-bit value that is added to the alpha value for each frame. The MSB (bit 15) indicates the sign (i.e., increment or decrement). When this value reaches either the maximum or the minimum alpha value (255 or 0) it keeps that value (i.e., it is not incremented/decremented) until it is reloaded via bit 17 (LOAD_ALPHA).
7:0	**ALPHA1_VAL (Alpha Window 1 Value).** Specifies the alpha value to be used for this window.
Offset 70h-73h	**Alpha Window 2 X Position Register (R/W)** — **Reset Value: 00000000h**
Note:	H_TOTAL and H_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8330h[26:19] and 8338h[10:3], respectively). The value of (H_TOTAL – H_SYNC_END) is sometimes referred to as "horizontal back porch". For more information, see the *GX1 Processor Series Datasheet*. Desired screen position should not be outside a video window (F4BAR0+Memory Offset 08h and 0Ch).
31:27	**Reserved.**
26:16	**ALPHA2_X_END (Alpha Window 2 Horizontal End).** Determines the horizontal end position of Alpha Window 2 (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 1.
15:11	**Reserved.**
10:0	**ALPHA2_X_START (Alpha Window 2 Horizontal Start).** Determines the horizontal start position of Alpha Window 2. This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 2.
Offset 74h-77h	**Alpha Window 2 Y Position Register (R/W)** — **Reset Value: 00000000h**
Note:	V_TOTAL and V_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8340h[26:16] and 8348h[26:16], respectively). The value of (V_TOTAL – V_SYNC_END) is sometimes referred to as "vertical back porch". For more information, see the *GX1 Processor Series Datasheet*. Desired screen position should not be outside a video window (F4BAR0+Memory Offset 08h and 0Ch).
31:27	**Reserved.**
26:16	**ALPHA2_Y_END (Alpha Window 2 Vertical End).** Determines the vertical end position of Alpha Window 2 (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 2.
15:11	**Reserved.**
10:0	**ALPHA2_Y_START (Alpha Window 2 Vertical Start).** Determines the vertical start position of Alpha Window 2. This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 1.
Offset 78h-7Bh	**Alpha Window 2 Color Register (R/W)** — **Reset Value: 00000000h**
31:25	**Reserved.**
24	**ALPHA2_COLOR_REG_EN (Alpha Window 2 Color Register Enable).** Enable bit for the color key matching in Alpha Window 2. 0: Disable. Where there is a color key match, graphics and video are alpha-blended. 1: Enable. If this bit is enabled and the alpha window is enabled, then where there is a color key match, the color value (in bits [23:0], ALPHA2_COLOR_REG) is displayed.
23:0	**ALPHA2_COLOR_REG (Alpha Window 1 Color Register).** Specifies the color to be displayed inside Alpha Window 2 when there is a color key match in the alpha window. This is an RGB value (for RGB blending) or a YUV value (for YUV blending). In Interlaced YUV blending mode, Y/2 value should be used. This color is only displayed if the alpha window is enabled and bit 24 (ALPHA2_COLOR_REG_EN) is enabled.

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
Offset 7Ch-7Fh	**Alpha Window 2 Control Register (R/W)** — **Reset Value: 00000000h**
31:18	**Reserved.**
17	**LOAD_ALPHA (Load Alpha Value). (Write Only)** When set to 1, this bit causes the Video Processor to load the alpha value (in bits [7:0], ALPHA2_VAL) at the start of the next frame.
16	**ALPHA2_WIN_EN (Alpha Window 2 Enable).** Enable bit for Alpha Window 2. 0: Disable Alpha Window 2. 1: Enable Alpha Window 2. **Note:** Valid only if video window is enabled (F4BAR0+Memory Offset 00h[0] = 1).
15:8	**ALPHA2_INCR (Alpha Window 2 Increment).** Specifies the alpha value increment/decrement. This is a signed 8-bit value that is added to the alpha value for each frame. The MSB (bit 15) indicates the sign (i.e., increment or decrement). When this value reaches either the maximum or the minimum alpha value (255 or 0) it keeps that value (i.e., it is not incremented/decremented) until it is reloaded via bit 17 (LOAD_ALPHA).
7:0	**ALPHA2_VAL (Alpha Window 1 Value).** Specifies the alpha value to be used for this window.
Offset 80h-83h	**Alpha Window 3 X Position Register (R/W)** — **Reset Value: 00000000h** **Note:** H_TOTAL and H_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8330h[26:19] and 8338h[10:3], respectively). The value of (H_TOTAL – H_SYNC_END) is sometimes referred to as "horizontal back porch". For more information, see the *GX1 Processor Series Datasheet*. **Note:** Desired screen position should not be outside a video window (F4BAR0+Memory Offset 08h and 0Ch).
31:27	**Reserved.**
26:16	**ALPHA3_X_END (Alpha Window 3 Horizontal End).** Determines the horizontal end position of Alpha Window 3 (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 1.
15:11	**Reserved.**
10:0	**ALPHA3_X_START (Alpha Window 3 Horizontal Start).** Determines the horizontal start position of Alpha Window 3. This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 2.
Offset 84h-87h	**Alpha Window 3 Y Position Register (R/W)** — **Reset Value: 00000000h** **Note:** V_TOTAL and V_SYNC_END are values programmed in the GX1 module's Display Controller Timing registers (GX_BASE+Memory Offset 8340h[26:16] and 8348h[26:16], respectively). The value of (V_TOTAL – V_SYNC_END) is sometimes referred to as "vertical back porch". For more information, see the *GX1 Processor Series Datasheet*. Desired screen position should not be outside a video window (F4BAR0+Memory Offset 08h and 0Ch).
31:27	**Reserved.**
26:16	**ALPHA3_Y_END (Alpha Window 3 Vertical End).** Determines the vertical end position of Alpha Window 3 (not inclusive). This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 2.
15:11	**Reserved.**
10:0	**ALPHA3_Y_START (Alpha Window 3 Vertical End).** Determines the vertical start position of Alpha Window 3. This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 1.
Offset 88h-8Bh	**Alpha Window 3 Color Register (R/W)** — **Reset Value: 00000000h**
31:25	**Reserved.**
24	**ALPHA3_COLOR_REG_EN (Alpha Window 3 Color Register Enable).** Enable bit for the color key matching in Alpha Window 3. 0: Disable. Where there is a color key match, graphics and video are alpha-blended. 1: Enable. If this bit is enabled and the alpha window is enabled, then where there is a color key match, the color value (in bits [23:0], ALPHA3_COLOR_REG) is displayed.
23:0	**ALPHA3_COLOR_REG (Alpha Window 3 Color Register).** Specifies the color to be displayed inside Alpha Window 3 when there is a color key match in the alpha window. This is an RGB value (for RGB blending) or a YUV value (for YUV blending). In Interlaced YUV blending mode, Y/2 value should be used. This color is only displayed if the alpha window is enabled and the bit 24 (ALPHA3_COLOR_REG_EN) is enabled.

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
Offset 8Ch-8Fh	**Alpha Window 3 Control Register (R/W)** — **Reset Value: 00000000h**
31:18	**Reserved.**
17	**LOAD_ALPHA (Load Alpha Value). (Write Only)** When set to 1, this bit causes the Video Processor to load the alpha value (in bits [7:0], ALPHA3_VAL) at the start of the next frame.
16	**ALPHA3_WIN_EN (Alpha Window 3 Enable).** Enable bit for Alpha Window 3. 0: Disable Alpha Window 3. 1: Enable Alpha Window 3. Valid only if video window is enabled (F4BAR0+Memory Offset 00h[0] = 1)
15:8	**ALPHA3_INCR (Alpha Window 3 Increment).** Specifies the alpha value increment/decrement. This is a signed 8-bit value that is added to the alpha value for each frame. The MSB (bit 15) indicates the sign (i.e., increment or decrement). When this value reaches either the maximum or the minimum alpha value (255 or 0) it keeps that value (i.e., it is not incremented/decremented) until it is reloaded via bit 17 (LOAD_ALPHA).
7:0	**ALPHA3_VAL (Alpha Window 3 Value).** Specifies the alpha value to be used for this window.
Offset 90h-93h	**Video Request Register (R/W)** — **Reset Value: 001B0017h**
31:28	**Reserved.** Set to 0.
27:16	**VIDEO_X_REQ (Video Horizontal Request).** Determines the horizontal (pixel) location at which to start requesting video data out of the video FIFO. This value is calculated according to the following formula: Value = Desired screen position + (H_TOTAL – H_SYNC_END) – 2.
15:11	**Reserved**.
10:0	**VIDEO_Y_REQ (Video Vertical Request).** Determines the line number at which to start requesting video data out of the video FIFO. This value is calculated according to the following formula: Value = Desired screen position + (V_TOTAL – V_SYNC_END) + 1.
Offset 94h-97h	**Alpha Watch Register (RO)** — **Reset Value: 00000000h** Alpha values may be automatically incremented/decremented for successive frames. This register can be used to read the alpha values that are being used in the current frame.
31:24	**Reserved.**
23:16	**ALPHA3_VAL (Value for Alpha Window 3).**
15:8	**ALPHA2_VAL (Value for Alpha Window 2).**
7:0	**ALPHA1_VAL (Value for Alpha Window 1).**
Offset 98h-3FFh	**Reserved**
Offset 400h-403h	**Video Processor Display Mode Register (R/W)** — **Reset Value: 00000000h** Selects various Video Processor modes.
31	**Video FIFO Underflow (Empty).** 0: No underflow has occurred. 1: Underflow has occurred. Write 1 to reset this bit.
30	**Video FIFO OverFlow (Full).** 0: No overflow has occurred. 1: Overflow has occurred. Write 1 to reset this bit.
29	**Reserved.** Write as read.
28	**Reserved.** Write as read.
27:4	**Reserved**. Set to 0.
3	**Reserved.** Write as read.
2	**Reserved.** Write as read.

Video Processor Module (Continued)

Table 6-7. F4BAR0+Memory Offset: Video Processor Configuration Registers (Continued)

Bit	Description
1:0	**VID_SEL (Video Select).** Selects the source of the video data. 00: GX1 module. 10: VIP block. 01: Reserved. 11: Reserved. The GX1 module's video clock must be active at all times, regardless of the source of video input.
Offset 404h-407h	**Reserved** **Reset Value: 00000000h**
Offset 408h-40Bh	**Video Processor Test Mode Register (R/W)** **Reset Value: 00000000h**
31:0	**Reserved.**
Offset 40Ch-41Fh	**Reserved**
Offset 420h-423h	**GenLock Register (R/W)** **Reset Value: 00000000h**
31:24	**Reserved.** Must be set to 0.
23	**0DD_TO (Odd Field Time Out).** Indicates CGENTO0 (F4BAR0+Memory Offset 43Ch[15:0]) has expired. This bit can be reset by writing 1 to it.
22	**EVEN_TO (Even Field Time Out).** Indicates CGENTO1 (F4BAR0+Memory Offset 43Ch[31:16]) has expired. This bit can be reset by writing 1 to it.
21:9	**Reserved.**
8	**Reserved.** Set to 0.
7	**Reserved.** Set to 0.
6	**Reserved.** Set to 0.
5	**Reserved.** Set to 0.
4	**GENLOCK_TOUT_EN (GenLock Timeout Enable).** 0: Disable. 1: Enable timeout.
3	**VIP_VSYNC_EDGE_SEL (VIP VSYNC Edge Select).** Selects which edge of the VSYNC signal should be synchronized with VIP. 0: Rising edge. 1: Falling edge.
2	**GX1_VSYNC_EDGE_SEL (GX1 VSYNC Edge Select).** Selects which edge of the VSYNC signal should be synchronized with the GX1 module. 0: Rising edge. 1: Falling edge.
1	**CT_GENLOCK_EN (Enable Continuous GenLock Function).** 0: The continuous GenLock function is disabled. 1: Enable locking (i.e., synchronization) of the GX1 VSYNC with the VIP VSYNC on every VSYNC (i.e., continuous locking). **Note:** If bit 0 (SG_GENLOCK_EN) = 1, it overrides the value of this bit.
0	**Reserved.** Set to 0.
Offset 424h-427h	**GenLock Delay Register (R/W)** **Reset Value: 00000000h**
31:21	**Reserved.**
20:0	**GENLOCK_DEL (GenLock Delay).** Indicates the delay (in 27 MHz clocks) between the VIP VSYNC and the GX1 module's Display Controller VSYNC.
Offset 428h-43Bh	**Reserved**
Offset 43Ch-43Fh	**Continuous GenLock Timeout Register (R/W)** **Reset Value: 1FFF1FFFh**
31:16	**CGENTO1 (Even Field Continuous GenLock Timeout).**
15:0	**CGENTO0 (Odd Field Continuous GenLock Timeout).**

Video Processor Module (Continued)

6.3.2.2 VIP Support Registers - F4BAR2

F4 Index 18h, Base Address Register 2 (F4BAR2) points to the base address of where the VIP Configuration registers are located. Table 6-8 shows the memory mapped VIP support registers accessed through F4BAR2.

Table 6-8. F4BAR2+Memory Offset: VIP Configuration Registers

Bit	Description
Offset 00h-03h	**Video Interface Port Configuration Register (R/W)** — **Reset Value: 00000000h**
31:23	**Reserved.** Must be set to 0.
22	**VIP FIFO Bus Request Threshold.** VIP FIFO issues a bus request when it is filled with 32 or 64 bytes. 0: 64 bytes. 1: 32 bytes
21	**VBI Task B Store to Memory.** When this bit is enabled, VBI task B data is stored to memory. 0: Disable. 1: Enable. This bit is relevant only if bit 18 (VBI Configuration Override) = 1 (enabled).
20	**VBI Task A Store to Memory.** When this bit is enabled, VBI task A data is stored to memory. 0: Disable. 1: Enable. This bit is relevant only if bit 18 (VBI Configuration Override) = 1 (enabled).
19	**VBI Ancillary Store to Memory.** When this bit is enabled, VBI Ancillary data is stored to memory. 0: Disable. 1: Enable. This bit is relevant only if bit 18 (VBI Configuration Override) = 1 (enabled).
18	**VBI Configuration Override.** When this bit is enabled, bits [21:19] override the setup specified in bits 17 and 16. 0: Disable. 1: Enable.
17	**VBI Data Task.** Specifies the CCIR656 video stream task used to store VBI data to memory. 0: Task B. 1: Task A. This bit is relevant only if bit 16 (VBI Mode for CCIR656) = 1 and bit 18 (VBI Configuration Override) = 0 (disabled).
16	**VBI Mode for CCIR656.** Specifies the mode in which to store VBI data to memory. 0: Use CCIR656 ancillary data to store VBI data to memory. 1: Use CCIR656 video task A or B to store VBI data to memory, depending on the value of bit 17 (VBI Task). This bit is only used if bit 18 (VBI Configuration Override) = 0 (disabled).
15:2	**Reserved.** Set to 0.
1:0	**Video Input Port Mode.** Selects VIP operating mode. 10: CCIR656 mode. All other decodes: Reserved.
Offset 04h-07h	**Video Interface Control Register (R/W)** — **Reset Value: 00000000h**
31:18	**Reserved.** Must be set to 0.
17	**Line Interrupt.** When asserted, allows interrupt (INTC#) generation when the Video Current Line register (F4BAR2+ Memory Offset 10h) contents equal the Video Line Target Register (F4BAR2+ Memory Offset 14h) contents. 0: Disable. 1: Enable.
16	**Field Interrupt.** When asserted, allows interrupt (INTC#) generation at the end of a field (i.e., the end of active video for the current field). Interrupt generation can be enabled regardless of whether or not video capture (store to memory) is enabled. 0: Disable. 1: Enable.
15:11	**Reserved.** Must be set to 0.

Video Processor Module (Continued)

Table 6-8. F4BAR2+Memory Offset: VIP Configuration Registers (Continued)

Bit	Description
10	**Auto-Flip.** Video port operation mode. 0: The video port automatically detects the even and odd fields based on the VP_HREF and VP_VSYNC_IN signals or the CCIR656 control codes. 1: The even/odd field detect logic is disabled and the video port automatically toggles between the even and odd buffers during capture. The odd buffer is the first to be filled in this mode. This bit must be programmed to 0 when Direct Video mode is used. Direct Video mode is used when VID_SEL = 10 (F4BAR0+Memory Offset 400h[1:0]). Otherwise the video select from the GX1 module. VID_SEL indicates the source of the video data.
9	**Capture (Store to Memory) VBI Data.** 0: Disable. 1: Enable.
8	**Capture (Store to Memory) Video Data.** 0: Disable. 1: Enable.
7:2	**Reserved.** Must be set to 0.
1:0	**Run Mode Capture.** Selects capture run mode. 00: Stop capture at end of current line. 01: Stop capture at end of current field. 10 Reserved. 11: Start capture at beginning of next field.
Offset 08h-0Bh	**Video Interface Status Register (R/W)** — **Reset Value: xxxxxxxxh**
31:25	**Reserved.(Read Only)**
24	**Current Field. (Read Only)** 0: Even field is being processed. 1: Odd field is being processed.
23:22	**Reserved. (Read Only)**
21	**Base Register Not Updated. (Read Only)** When set to 1, this bit indicates that one of the base registers (at F4BAR2+Memory Offset 20h, 24h, 40h, and 44h) has been written but has not yet been updated. 0: All base registers are updated. 1: One or more of the base registers has not been updated.
20	**FIFO Overflow Status Indication.** 0: No overflow occurred. 1: An overflow occurred for the FIFO between the VIP and the Fast X-Bus. Writing a 1 to this bit clears the status.
19:18	**Reserved. (Read Only)**
17	**Line Interrupt (INTC#) Pending Status.** 0: Interrupt not pending. 1: Interrupt pending. Writing a 1 to this bit clears the status.
16	**Field Interrupt (INTC) Pending Status.** 0: Interrupt not pending. 1: Interrupt pending. Writing a 1 to this bit clears the status.
15:10	**Reserved. (Read Only)**
9	**VBI Data Capture Active. (Read Only)** 0: VBI data is not being stored to memory. 1: VBI data is now being stored to memory.
8	**Video Data Capture Active. (Read Only)** 0: Video data is not being stored to memory. 1: Video data is now being stored to memory.

Table 6-8. F4BAR2+Memory Offset: VIP Configuration Registers (Continued)

Bit	Description
7:1	**Reserved. (Read Only)**
0	**Run Status. (Read Only)** 0: Video port capture is not active. 1: Video port capture is in progress.
Offset 0Ch-0Fh	**Reserved** — **Reset Value: 00h**
Offset 10h-13h	**Video Current Line Register (RO)** — **Reset Value: xxxxxxxxh**
31:10	**Reserved.**
9:0	**Current Line.** Indicates the video line currently being stored to memory. The count indicated in this field is reset to 0 at the start of each field.
Offset 14h-17h	**Video Line Target Register (R/W)** — **Reset Value: 00000000h**
31:10	**Reserved.** Must be set to 0.
9:0	**Line Target.** Indicates the video line to generate an interrupt on.
Offset 18h-1Bh	**Reserved** — **Reset Value: 00000000h**
Offset 1Ch-1Fh	**Reserved** — **Reset Value: 00000000h**
Offset 20h-23h	**Video Data Odd Base Register (R/W)** — **Reset Value: 00000000h**
This register specifies the base address in graphics memory where odd video field data are stored. Changes to this register take effect at the beginning of the next field. The value in this register is 16-byte aligned. **Note:** This register is double-buffered. When a new value is written to this register, the new value is placed in a special "pending" register, and the "Base Register Not Updated" bit (F4BAR2+MemoryOffset 08h[21]) is set to 1. The Video Data Odd Base register (this register) is not updated at this point. When the first data of the next field is stored to memory, the pending values of all base registers (including this one) are written to the appropriate base registers, and the "Base Register Not Updated" bit is cleared.	
31:0	**Video Odd Base Address.** Base address where odd video data are stored in graphics memory. Bits [3:0] are always 0, and define the required address space.
Offset 24h-27h	**Video Data Even Base Register (R/W)** — **Reset Value: 00000000h**
This register specifies the base address in graphics memory where even video field data are stored. Changes to this register take effect at the beginning of the next field. The value in this register is 16-byte aligned. **Note:** This register is double-buffered. When a new value is written to this register, the new value is placed in a special "pending" register, and the "Base Register Not Updated" bit (F4BAR2+MemoryOffset 08h[21]) is set to 1. The Video Data Even Base register (this register) is not updated at this point. When the first data of the next field is stored to memory, the pending values of all base registers (including this one) are written to the appropriate base registers, and the "Base Register Not Updated" bit is cleared.	
31:0	**Video Even Base Address.** Base address where even video data are stored in graphics memory. Bits [3:0] are always 0, and define the required address space.
Offset 28h-2Bh	**Video Data Pitch Register (R/W)** — **Reset Value: 00000000h**
This register specifies the logical width of the video data buffer. This value is added to the start of the line address to get the address of the next line where video data are stored to memory. This value must be an integral number of DWORDs.	
31:16	**Reserved.**
15:0	**Video Data Pitch.** Specifies the logical width of the video data buffer. Bits [1:0] are always 0.
Offset 2Ch-3Fh	**Reserved** — **Reset Value: 00000000h**
Offset 40h-43h	**VBI Data Odd Base Register (R/W)** — **Reset Value: 00000000h**
This register specifies the base address in graphics memory where VBI data for odd fields are stored. Changes to this register take effect at the beginning of the next field. The value in this register is 16-byte aligned. **Note:** This register is double-buffered. When a new value is written this register, the new value is placed in a special "pending" register, and the "Base Register Not Updated" bit (F4BAR2+MemoryOffset 08h[21]) is set to 1. The VBI Data Odd Base Register (this register) is not updated at this point. When the first data of the next field is stored to memory, the pending values of all base registers (including this one) are written to the appropriate base registers, and the "Base Register Not Updated" bit is cleared.	
31:0	**VBI Odd Base Address.** Base address where VBI data for odd fields is stored in graphics memory. Bits [3:0] are always 0 and define the required address space.

Video Processor Module (Continued)

Table 6-8. F4BAR2+Memory Offset: VIP Configuration Registers (Continued)

Bit	Description
Offset 44h-47h	**VBI Data Even Base Register (R/W)** — **Reset Value: 00000000h**
This register specifies the base address in graphics memory where VBI data for even fields is stored. Changes to this register take effect at the beginning of the next field. The value in this register is 16-byte aligned. **Note:** This register is double-buffered. When a new value is written to this register, the new value is placed in a special "pending" register, and the "Base Register Not Updated" bit (F4BAR2+MemoryOffset 08h[21]) is set to 1. The VBI Data Even Base Register (this register) is not updated at this point. When the first data of the next field is stored to memory, the pending values of all base registers (including this one) are written to the appropriate base registers, and the "Base Register Not Updated" bit is cleared.	
31:0	**VBI Even Base Address.** Base address where VBI data for even fields is stored in graphics memory. Bits [3:0] are always 0 and define the required address space.
Offset 48h-4Bh	**VBI Data Pitch Register (R/W)** — **Reset Value: 00000000h**
This register specifies the logical width of the VBI data buffer. This value is added to the start of the line address to get the address of the next line where VBI data are stored to memory. This value must be an integral number of DWORDs.	
31:16	**Reserved.**
15:0	**VBI Data Pitch.** Specifies the logical width of the video data buffer. Bits [1:0] are always 0.
Offset 4Ch-1FFh	**Reserved** — **Reset Value: 00h**

7.0 Debugging and Monitoring

7.1 TESTABILITY (JTAG)

The Test Access Port (TAP) allows board level interconnection verification and chip production tests. An IEEE-1149.1a compliant test interface, TAP supports all IEEE mandatory instructions as well as several optional instructions for added functionality. See Table 7-1 for a summary of all instructions support. For further information on JTAG, refer to IEEE Standard 1149.1a-1993 Test Access Port and Boundary-Scan Architecture.

7.1.1 Mandatory Instruction Support

The TAP supports all IEEE mandatory instructions, including:

- BYPASS
 Presents the shortest path through a given chip (a 1-bit shift register).
- EXTEST
 Drives data loaded into the JTAG path (possibly with a SAMPLE/PRELOAD instruction) to output pins.
- SAMPLE/PRELOAD
 Captures chip inputs and outputs.

7.1.2 Optional Instruction Support

The TAP supports the following IEEE optional instructions:

- IDCODE
 Presents the contents of the Device Identification register in serial format.
- CLAMP
 Ensures that the Bypass register is connected between TDI and TDO, and then drives data that was loaded into the Boundary Scan register (e.g., via SAMPLE-PRELOAD instruction) to output signals. These signals do not change while the CLAMP instruction is selected.
- HIZ
 Puts all chip outputs in inactive (floating) state (including all pins that do not require a TRI-STATE output for normal functionality). Note that not all pull-up resistors are disabled in this state.

7.1.3 JTAG Chain

Balls that are not part of the JTAG chain:

- CRT DACs
- USB I/Os

Table 7-1. JTAG Mode Instruction Support

Code	Instruction	Activity
000	EXTEST	Drives shifted data to output pins.
001	SAMPLE/PRELOAD	Captures inputs and system outputs.
010	IDCODE	Scans out device identifier.
011	HIZ	Puts all output and bidirectional pins in TRI-STATE mode.
100	CLAMP	Drives fixed data from Boundary Scan register.
101	Reserved	
110	Reserved	
111	BYPASS	Presents shortest external path through device.

8.0 Electrical Specifications

This chapter provides information about:

- General electrical specifications.
- DC characteristics.
- AC characteristics.
- All voltage values in this chapter are with respect to V_{SS} unless otherwise noted.

8.1 GENERAL SPECIFICATIONS

8.1.1 Power/Ground Connections and Decoupling

When testing and operating the SC2200, use standard high frequency techniques to reduce parasitic effects. For example:

- Filter the DC power leads with low-inductance decoupling capacitors.
- Use low-impedance wiring.
- Utilizing the PWR and GND pins.

8.1.2 Absolute Maximum Ratings

Stresses beyond those indicated in the following table may cause permanent damage to the SC2200, reduce device reliability and result in premature failure, even when there is no immediately apparent sign of failure. Prolonged exposure to conditions at or near the absolute maximum ratings may also result in reduced device life span and reduced reliability.

Note: The values in the following table are stress ratings only. They do not imply that operation under other conditions is impossible.

8.1.3 Operating Conditions

Table 8-2 lists the various power supplies of the SC2200 and provides the device operating conditions.

Table 8-1. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit	Comments
T_{CASE}	Operating case temperature[1]	-45	110	^oC	
$T_{STORAGE}$	Storage temperature[2]	-45	125	^oC	
V_{CC}	Supply voltage		See Table 8-2	V	
V_{MAX}	Voltage on				
	5V tolerant balls[3]	-0.5	6.0	V	
	Others[3,4]	-0.5	3.6	V	
I_{IK}	Input clamp current[1]	-0.5	10	mA	
I_{OK}	Output clamp current[1]		25	mA	

1. Power applied - no clocks.
2. No bias.
3. Voltage min is -0.8V with a transient voltage of 20 ns or less.
4. Voltage max is 4.0V with a transient voltage of 20 ns or less.

Electrical Specifications (Continued)

Table 8-2. Operating Conditions

Symbol[1]	Parameter	Min	Typ	Max	Unit	Comments
T_C	Operating case temperature	0	-	85	°C	
AV_{CCUSB} AV_{CCCRT}	Analog power supply. Powers internal analog circuits and some external signals (see Table 8-3).	3.14	3.3	3.46	V	
V_{BAT}	Battery supply voltage. Powers RTC and ACPI when V_{BAT} is greater than V_{SB} (by at least 0.5V), and some external signals (see Table 8-3).	2.4	3.0	3.46	V	
V_{IO}	I/O buffer power supply. Powers most of the external signals (see Table 8-3); certain signals within this power plane are 5V tolerant.	3.14	3.3	3.46	V	
V_{CORE}	Core processor and internal digital power supply. Powers internal digital logic, including internal frequency multipliers.					
	233 or 266 MHz	1.71	1.8	1.89	V	
	300 MHz	1.99	2.1	2.21		
V_{PLL2} V_{PLL3}	PLL. Internal Phase Locked Loops (PLLs) power supply.	3.14	3.3	3.46	V	
V_{SB}	Standby power supply. Powers RTC and ACPI when V_{SB} is greater than V_{BAT}-0.5V, and some external signals (see Table 8-3).	3.14	3.3	3.46	V	
V_{SBL}	Standby logic. Powers internal logic needed to support Standby V_{SB}. V_{SBL} requires a 0.1 µF bypass capacitor to V_{SS}.					
	233 or 266 MHz	1.71	1.8	1.89	V	
	300 MHz	1.99	2.1	2.21		
V_{CCCRT}	CRT DAC. Powers CRT DAC digital circuits.					
	233 or 266 MHz	1.71	1.8	1.89	V	
	300 MHz	1.99	2.1	2.21		

1. For V_{IH} (Input High Voltage), V_{IL} (Input Low Voltage), I_{OH} (Output High Current), and I_{OL} (Output Low Current) operating conditions refer to Section 8.2 "DC Characteristics" on page 365.

Notes:

1) All power sources must be connected to the SC2200, even if the function is not used.

2) V_{SB}, V_{SBL} and V_{BAT} must be on if any other voltage is applied. V_{SB} and V_{BAT} voltages can be applied separately. See Section 8.3.15 "Power-Up Sequencing" on page 425.

3) The power planes of the SC2200 can be turned on or off. For more information, see Section 5.2.9 "Power Management Logic" on page 167.

4) It is recommended that the voltage difference between V_{CCCRT}, V_{CORE} and V_{SBL} be less than 0.25V, in order to reduce leakage current. If the voltage difference exceeds 0.25V, excessive leakage current is used in gates that are connected on the boundary between voltage domains.

5) It is recommended that the voltage difference between V_{IO} and V_{SB} be less than 0.25V, in order to reduce leakage current. If the voltage difference exceeds 0.25V, excessive leakage current is used in gates that are connected on the boundary between voltage domains.

Electrical Specifications (Continued)

Table 8-3 indicates which power rails are used for each signal of the SC2200 external interface. Power planes not listed in this table are internal, and are not related to signals of the external interface.

Table 8-3. Power Planes of External Interface Signals

Power Plane	Signal Names	V_{CC} Balls	V_{SS} Balls
Standby	GPWIO[0:2], LED#, ONCTL#, PWRBTN#, PWRCNT[1:2], THRM#, CLK32, IRRX1, RI2#, SDATA_IN2	V_{SB}	V_{SS}
Battery	X32I, X32O	V_{BAT}	V_{SS}
CRT DAC	RED, GREEN, BLUE, VREF, SETRES	AV_{CCCRT}	AV_{SSCRT}
USB	DPOS_PORT1, DNEG_PORT1, DPOS_PORT2, DNEG_PORT2, DPOS_PORT3, DNEG_PORT3	AV_{CCUSB}	AV_{SSUSB}
I/O	All other external interface signals	V_{IO}	V_{SS}

8.1.4 DC Current

DC current is not a simple measurement. Three of the SC2200 power states (On, Active Idle, Sleep) were selected for measurement. For each power state measured, two functional characteristics (Typical Average, Absolute Maximum) are used to determine how much current the SC2200 uses.

8.1.4.1 Power State Parameter Definitions

The DC characteristics tables in this section list Core and I/O current for three of the power states. For more explanation on the SC2200 power states see Section 5.2.9 "Power Management Logic" on page 167.

- **On (C0)**: All internal and external clocks with respect to the SC2200 are running and all functional blocks inside the GX1 module (CPU Core, Memory Controller, Display Controller, etc.) are actively generating cycles. This is equivalent to the ACPI specification's “S0,C0” state.
- **Active Idle (C1)**: The CPU Core has been halted, all other functional blocks (including the Display Controller for refreshing the display) are actively generating cycles. This state is entered when a HLT instruction is executed by the CPU Core. From a user's perspective, this state is indistinguishable from the On state and is equivalent to the ACPI specification's “S0,C1” state.
- **Sleep (SL2)**: This is the lowest power state the SC2200 can be in with voltage still applied to the device's core and I/O supply pins. This is equivalent to the ACPI specification's “S1” state.

8.1.4.2 Definition and Measurement Techniques of SC2200 Current Parameters

The following two parameters describes the SC2200 current while in the On state:

- **Typical Average**: Indicates the average current used by the SC2200 while in the On state. This is measured by running typical Windows applications in a typical display mode. In this case, 800x600x8 bpp at 75 Hz, 50 MHz DCLK using a background image of vertical stripes (4-pixel wide) alternating between black and white with power management disabled (to guarantee that the SC2200 never goes into the Active Idle state). This number is provided for reference only since it can vary greatly depending on the usage model of the system.

Note: This typical average should not be confused with the typical power numbers. Typical power is based on a combination of On (Typical Average) and Active Idle states.

- **Absolute Maximum**: Indicates the maximum instantaneous current used by the SC2200. CPU Core current is measured by running the Landmark Speed 200 benchmark test (with power management disabled) and measuring the peak current at any given instant during the test. I/O current is measured by running Microsoft Windows 98 and using a background image of vertical stripes (1-pixel wide) alternating between black and white at the maximum display resolution of each of the display type supported (CRT and TFT).

Electrical Specifications (Continued)

8.1.4.3 Definition of System Conditions for Measuring On Parameters

The SC2200's current is highly dependent on two functional characteristics, DCLK (DOT clock) and SDRAM frequency. Table 8-4 shows how these factors are controlled when measuring the typical average and absolute maximum processor current parameters.

8.1.4.4 DC Current Measurements

Table 8-5 and Table 8-6 show the DC current measurements of the SC2200. The SC2200 supports CRT and TFT displays, but it is expected that generally only one display interface will be used. Power consumed by the SC2200 is different with different displays. The CRT DAC requires current, while the TFT interface even though it has no DAC to power, also draws current while it is active. The CRT DAC and the TFT interface are presented as separate line items. The chosen display type I/O current should be added to the Typical, Absolute Maximum, and Active Idle I/O currents to get total current.

Table 8-4. System Conditions Used to Measure SC2200's Current Used During the "On" State

CPU Current Measurement	System Conditions			
	V_{CORE}[1]	V_{IO}[1]	DCLK Freq.	SDRAM Freq.
Typical Average	Nominal	Nominal	50 MHz[2]	Nominal
Absolute Maximum	Max	Max	135 MHz[3]	Max

1. See Table 8-2 on page 359 for nominal and maximum voltages.
2. A DCLK frequency of 50 MHz is derived by setting the display mode to 800x600x8 bpp at 75 Hz, using a display image of vertical stripes (4-pixel wide) alternating between black and white with power management disabled.
3. A DCLK frequency of 135 MHz is derived by setting the display mode to 1280x1024x8 bpp at 75 Hz, using a display image of vertical stripes (1-pixel wide) alternating between black and white with power management disabled.

Table 8-5. DC Characteristics On Mode

Symbol	Parameter[1]	Typ Avg	Abs Max	Unit	Comments
I_{CC3ON}	f_{CLK} = 233 MHz, I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = On, excludes TFT interface contribution and CRT DAC	230	250	mA	I_{CC} for V_{IO}
	f_{CLK} = 266 MHz, I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = On, excludes TFT interface contribution and CRT DAC	240	260		
	f_{CLK} = 300 MHz, I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = On, excludes TFT interface contribution and CRT DAC	250	270		
I_{COREON}	f_{CLK} = 233 MHz, Core Current @ V_{CORE} = 1.8V (Nominal); CPU mode = On	820	990	mA	I_{CC} for V_{CORE}
	f_{CLK} = 266 MHz, Core Current @ V_{CORE} = 1.8V (Nominal); CPU mode = On	900	1090		
	f_{CLK} = 300 MHz, Core Current @ V_{CORE} = 2.1V (Nominal); CPU mode = On	1100	1400		
I_{SBON}	SB Current @ V_{SB} = 3.3V (Nominal); CPU mode = On	1	2	mA	
I_{SBLON}	SBL Current @ V_{SBL} = 1.8V (Nominal); CPU mode = On	10	20	mA	
	SBL Current @ V_{SBL} = 2.1V (Nominal); CPU mode = On	10	20		

Electrical Specifications (Continued)

Table 8-5. DC Characteristics On Mode (Continued)

Symbol	Parameter[1]	Typ Avg	Abs Max	Unit	Comments
$I_{CC3ONTFT}$	I/O current contribution if TFT display is used	30	50	mA	
I_{CCCRT}	If CRT interface is used: CCCRT Current @ V_{CCCRT} = 3.3 (Nominal); CPU mode = On	60	80	mA	

1. f_{CLK} ratings refer to internal clock frequency.

Table 8-6. DC Characteristics for Active Idle, Sleep, and Off Modes

Symbol	Parameter[1]	Min	Typ	Max	Unit	Comments
$I_{CC3IDLE}$	f_{CLK} = 233 MHz, I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = Active Idle		230		mA	I_{CC} for V_{IO}
	f_{CLK} = 266 MHz, I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = Active Idle		240			
	f_{CLK} = 300 MHz, I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = Active Idle		250			
I_{CC3SLP}	I/O Current @ V_{IO} = 3.3V (Nominal); CPU mode = Sleep		20	30	mA	I_{CC} for V_{IO}[2]
$I_{COREIDLE}$	f_{CLK} = 233 MHz, Core Current @ V_{CORE} = 1.8V (Nominal); CPU mode = Active Idle		360		mA	I_{CC} for V_{CORE}
	f_{CLK} = 266 MHz, Core Current @ V_{CORE} = 1.8V (Nominal); CPU mode = Active Idle		380			
	f_{CLK} = 300 MHz, Core Current @ V_{CORE} = 2.1V (Nominal); CPU mode = Active Idle		470			
$I_{CORESLP}$	Core Current @ V_{CORE} = 1.8V (Nominal); CPU mode = Sleep		20	30	mA	I_{CC} for V_{CORE}[2]
	Core Current @ V_{CORE} = 2.1V (Nominal); CPU mode = Sleep		20	30		
I_{SBOFF}	SB Current @ V_{SB} = 3.3V (Nominal); CPU mode = Off		<1		mA	
I_{SBLOFF}	SBL Current @ V_{SBL} = 1.8V (Nominal); CPU mode = Off		<1		mA	I_{CC} for V_{SBL}[3]
	SBL Current @ V_{SBL} = 2.1V (Nominal); CPU mode = Off		<1			
I_{BAT}[4]	BAT Current @ V_{BAT} = 3.0 (Nominal); CPU mode = Off		7	50	µA	T_C = 25°C
I_{BAT}[5]	BAT Current @ V_{BAT} = 3.0 (Nominal); CPU mode = Off		7	15	µA	T_C = 25°C

1. f_{CLK} ratings refer to internal clock frequency.
2. All inputs are at 0.2V or V_{IO} – 0.2 (CMOS levels). All inputs are held static, and all outputs are unloaded (static I_{OUT} = 0 mA).
3. All V_{SBL} supplied inputs are at 0.2V or V_{SBL} – 0.2 (CMOS levels). All inputs are held static, and all outputs are unloaded (static I_{OUT} = 0 mA).
4. Applies to NSIDs SC2200UCL-266, SC2200UCL-300, SC2200UFH-233, C2200UFH-266, and SC2200UFH300.
5. Applies to NSIDs SC2200UFH-233B and SC2200UFH-266B.

Electrical Specifications (Continued)

8.1.5 Ball Capacitance and Inductance

Table 8-7 gives ball capacitance and inductance values.

Table 8-7. Ball Capacitance and Inductance

Symbol	Parameter	Min	Typ	Max	Unit
C_{IN}[1]	Input Pin Capacitance		4	7	pF
C_{IN}[1]	Clock Input Capacitance	5	8	12	pF
C_{IO}[1]	I/O Pin Capacitance		10	12	pF
C_{O}[1]	Output Pin Capacitance		6	8	pF
L_{PIN}[2]	Pin Inductance			20	nH

1. T_A = 25°C, f = 1 MHz. All capacitances are not 100% tested.
2. Not 100% tested.

Electrical Specifications (Continued)

8.1.6 Pull-Up and Pull-Down Resistors

The following table lists input balls that are internally connected to a pull-up (PU) or pull-down (PD) resistor. If these balls are not used, they do not require connection to an external PU or PD resistor.

Note: The resistors described in this table are implemented as transistors. The resistance for PUs assumes $V_{IN} = V_{SS}$ and for PDs assumes $V_{IN} = V_{IO}$.

Table 8-8. Balls with PU/PD Resistors

Signal Name	Ball No. EBGA	Ball No. TEPBGA	PU/PD	Typ[1] Value [Ω]
PCI				
FRAME#	E1	D8	PU	22.5K
C/BE[3:0]#	A8, D8, A10, A13	H4, F3, J2, L1	PU	22.5K
PAR	C10	J4	PU	22.5K
IRDY#	C8	F2	PU	22.5K
TRDY#	B8	F1	PU	22.5K
STOP#	D9	G1	PU	22.5K
LOCK#	C9	H3	PU	22.5K
DEVSEL#	B5	E4	PU	22.5K
PERR#	B9	H2	PU	22.5K
SERR#	A9	H1	PU	22.5K
REQ[1:0]#	E3, C1	A5, B5	PU	22.5K
INTA#	AE3	D26	PU	22.5K
INTB#	AF1	C26	PU	22.5K
INTC#	H4	C9	PU	22.5K
INTD#	B22	AA2	PU	22.5K
Low Pin Count (LPC)				
LAD[3:0]	AJ10, AK10, AL10, AJ11	L29, L30, L31, M28	PU	22.5K
LDRQ	AL9	L28	PU	22.5K
SERIRQ	AL8	J31	PU	22.5K
System (Straps)				
CLKSEL[3:0]	AL13, AK3, B27, F3	P30, D29, AF3, B8	PD	100K
BOOT16	G4	C8	PD	100K
TFT_PRSNT	AK13	P29	PD	100K
LPC_ROM	E4	D6	PD	100K
FPCI_MON	D3	A4	PD	100K
DID[1:0]	D2, D4	C6, C5	PD	100K
ACCESS.bus[2]				
AB1C	AJ13	N31	PU	22.5K
AB1D	AL12	N30	PU	22.5K
AB2C	AJ12	N29	PU	22.5K
AB2D	AL11	M29	PU	22.5K
Parallel Port				
AFD#/DSTRB#	AB2	D22	PU	22.5K
PE	T3	D17	PU[2]	22.5K
			PD[2]	22.5K
SLIN#/ASTRB#	W1	B20	PU	22.5K
STB#/WRITE#	AB1	A22	PU	22.5K
INIT#	Y3	B21	PU	22.5K

Signal Name	Ball No. EBGA	Ball No. TEPBGA	PU/PD	Typ[1] Value [Ω]
JTAG				
TCK	AL4	E31	PU	22.5K
TMS	AJ5	F28	PU	22.5K
TDI	AK5	F29	PU	22.5K
TRST#	AK4	E29	PU	22.5K
GPIO[2]				
GPIO1	H2, AL12	D10, N30	PU	22.5K
GPIO6	AH3	D28	PU	22.5K
GPIO7	AH4	C30	PU	22.5K
GPIO8	AJ2	C31	PU	22.5K
GPIO9	AG4	C28	PU	22.5K
GPIO10	AJ1	B29	PU	22.5K
GPIO11	H30	AJ8	PU	22.5K
GPIO12	AJ12	N29	PU	22.5K
GPIO13	AL11	M29	PU	22.5K
GPIO14	F1	D9	PU	22.5K
GPIO15	G3	A8	PU	22.5K
GPIO16	AL15	V31	PU	22.5K
GPIO17	J4	A10	PU	22.5K
GPIO18	A28	AG1	PU	22.5K
GPIO19	H4	C9	PU	22.5K
GPIO20	H3, AJ13	A9, N31	PU	22.5K
GPIO32	AJ11	M28	PU	22.5K
GPIO33	AL10	L31	PU	22.5K
GPIO34	AK10	L30	PU	22.5K
GPIO35	AJ10	L29	PU	22.5K
GPIO36	AL9	L28	PU	22.5K
GPIO37	AK9	K31	PU	22.5K
GPIO38	AJ9	K28	PU	22.5K
GPIO39	AL8	J31	PU	22.5K
Power Management				
PWRBTN#	E29	AH5	PU	100K
GPWIO[2:0]	G29, G28, E31	AJ6, AK5, AH6	PU	100K
Test and Measurement				
GTEST	AL5	F30	PD	22.5K

1. Accuracy is: 22.5 KΩ resistors are within a range of 20 KΩ to 50 KΩ. 100 KΩ resistors are within a range of 90 KΩ to 250 KΩ.
2. Controlled by software.

Electrical Specifications (Continued)

8.2 DC CHARACTERISTICS

Table 8-9 describes the signal buffer types of the SC2200. See Table 2-2 "432-EBGA Ball Assignment - Sorted by Ball Number" on page 22 and Table 2-4 "481-TEPBGA Ball Assignment - Sorted by Ball Number" on page 37 for each signal's buffer type.

The subsections that follows provide detailed DC characteristics according to buffer type.

Table 8-9. Buffer Types

Symbol	Description	Reference
Diode	Diodes only, no buffer	---
IN_{AB}	Input, ACCESS.bus compatible with Schmitt Trigger	Section 8.2.1
IN_{BTN}	Input, TTL compatible with Schmitt Trigger, low leakage	Section 8.2.2
IN_{PCI}	Input, PCI compatible	Section 8.2.3
IN_{STRP}	Input, Strap ball (min V_{IH} is $0.6V_{IO}$) with weak pull-down	Section 8.2.4
IN_{T}	Input, TTL compatible	Section 8.2.5
IN_{TS}	Input, TTL compatible with Schmitt Trigger type 200 mV	Section 8.2.6
IN_{TS1}	Input, with Schmitt Trigger type 200 mV	Section 8.2.7
IN_{USB}	Input, USB compatible	Section 8.2.8
O_{AC97}	Output, Totem-Pole, AC97 compatible	Section 8.2.9
OD_{n}	Output, Open-Drain, capable of sinking *n* mA.[1]	Section 8.2.10
OD_{PCI}	Output, Open-Drain, PCI compatible	Section 8.2.11
$O_{p/n}$	Output, Totem-Pole, capable of sourcing *p* mA and sinking *n* mA	Section 8.2.12
O_{PCI}	Output, PCI compatible, TRI-STATE	Section 8.2.13
O_{USB}	Output, USB compatible	Section 8.2.14
$TS_{p/n}$	Output, TRI-STATE, capable of sourcing *p* mA and sinking *n* mA	Section 8.2.15
WIRE	Wire, no buffer	---

1. Output from these signals is open-drain and cannot be forced high.

Electrical Specifications (Continued)

8.2.1 IN_{AB} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	1.4		V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_{HIS}	Input hysteresis	150		mV	

1. Not 100% tested.

8.2.2 IN_{BTN} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	$V_{SB}+0.3$[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		5	µA	$V_{IN} = V_{SB}$
			-36	µA	$V_{IN} = V_{SS}$
V_{HIS}	Input Hysteresis[1]	250		mV	

1. Not 100% tested.

8.2.3 IN_{PCI} DC Characteristics

Note that the buffer type for PCICLK (EBGA ball E2 / TEPBGA ball A7) is IN_T - not IN_{PCI}.

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	$0.5V_{IO}$	$V_{IO}+0.3$[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	$0.3V_{IO}$	V	
V_{IPU}	Input Pull-up Voltage[2]	$0.7V_{IO}$		V	
I_{IL}	Input Leakage Current[3,4]		+/-10	µA	$0 < V_{IN} < V_{IO}$

1. Not 100% tested.
2. Not 100% tested. This parameter indicates the minimum voltage to which pull-up resistors are calculated in order to pull a floated network.
3. Input leakage currents include HIZ output leakage for all bidirectional buffers with TRI-STATE outputs.
4. See Exceptions 2 and 3 in Section 8.2.15.1 on page 369.

8.2.4 IN_{STRP} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	$0.6V_{IO}$	$V_{IO}+0.3$[1]	V	
V_{IL}	Input Low Voltage		$0.3V_{IO}$	V	
I_{IL}	Input Leakage Current		36	µA	During Reset: $V_{IN} = V_{IO}$
			–10	µA	$V_{IN} = V_{SS}$

1. Not 100% tested.

Electrical Specifications (Continued)

8.2.5 IN_T DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{IO}+0.3[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			–10	µA	$V_{IN} = V_{SS}$

1. Not 100% tested.

8.2.6 IN_{TS} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{IO}+0.3[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_H	Input Hysteresis	200		mV	

1. Not 100% tested.

8.2.7 IN_{TS1} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	$0.5V_{IO}$	V_{IO}+0.3[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	$0.3V_{IO}$	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_{HIS}	Input Hysteresis[1]	200		mV	

1. Not 100% tested.

8.2.8 IN_{USB} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{IO}+0.3[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_{DI}	Differential Input Sensitivity	0.2		V	\|(D+)-(D-)\| and Figure 8-1
V_{CM}	Differential Common Mode Range	0.8	2.5	V	Includes V_{DI} Range
V_{SE}	Single Ended Receiver Threshold	0.8	2.0	V	

1. Not 100% tested.

Electrical Specifications (Continued)

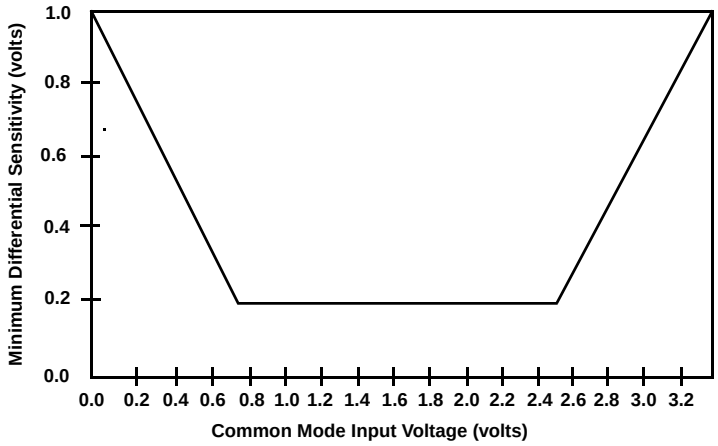


Figure 8-1. Differential Input Sensitivity for Common Mode Range

8.2.9 O_{AC97} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	$0.9V_{IO}$		V	I_{OH} = -5 mA
V_{OL}	Output Low Voltage		$0.1V_{IO}$	V	I_{OL} = 5 mA

8.2.10 OD_n DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = *n* mA

8.2.11 OD_{PCI} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OL}	Output Low Voltage		$0.1V_{IO}$	V	I_{OL} = 1500 µA

8.2.12 $O_{p/n}$ DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -*p* mA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = *n* mA

8.2.13 O_{PCI} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	$0.9V_{IO}$		V	I_{OH} = -500 µA
V_{OL}	Output Low Voltage		$0.1V_{IO}$	V	I_{OL} =1500 µA

Electrical Specifications (Continued)

8.2.14 O_{USB} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{USB_OH}	High-level Output Voltage	2.8	3.6[1]	V	I_{OH} = -0.25 mA R_L = 15 KΩ to GND
V_{USB_OL}	Low-level Output Voltage		0.3	V	I_{OL} = 2.5 mA R_L = 1.5 KΩ to 3.6V
t_{USB_CRS}	Output Signal Crossover Voltage	1.3	2.0	V	

1. Tested by characterization.

8.2.15 $TS_{p/n}$ DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	2.4		V	I_{OH} = *-p* mA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = *n* mA

8.2.15.1 Exceptions

1) I_{OH} is valid for a GPIO pin only when it is not configured as open-drain.
2) Signals with internal pull-ups have a maximum input leakage current of: $-\left(\frac{V_{power} - V_{IN}}{R(pull-up)}\right)$
 Where V_{power} is V_{IO}, or V_{SB}.
3) Signals with internal pull-downs have a maximum input leakage current of: $+\left(\frac{V_{IN} - V_{SS}}{R(pull-down)}\right)$

Electrical Specifications (Continued)

8.3 AC CHARACTERISTICS

The tables in this section list the following AC characteristics:

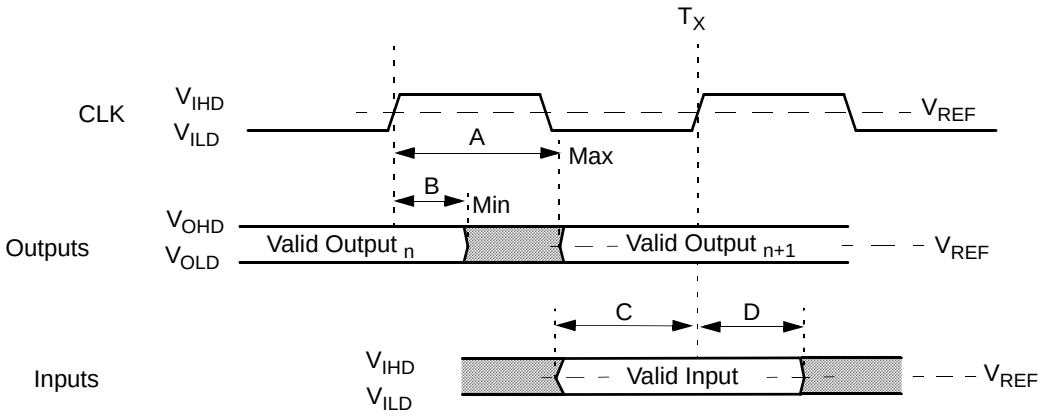
- Output delays
- Input setup requirements
- Input hold requirements
- Output float delays
- Power-up sequencing requirements

The default levels for measurement of the rising clock edge reference voltage (V_{REF}), and other voltages are shown in Table 8-10. Input or output signals must cross these levels during testing. Unless otherwise specified, all measurement points in this section conform to these default levels.

Table 8-10. Default Levels for Measurement of Switching Parameters

Symbol	Parameter	Value (V)
V_{REF}	Reference Voltage	1.5
V_{IHD}	Input High Drive Voltage	2.0
V_{ILD}	Input Low Drive Voltage	0.8
V_{OHD}	Output High Drive Voltage	2.4
V_{OLD}	Output Low Drive Voltage	0.4

All AC tests are at V_{IO} = 3.14V to 3.46V (3.3V nominal), T_C = 0 °C to 85 °C, C_L = 50 pF, unless otherwise specified.



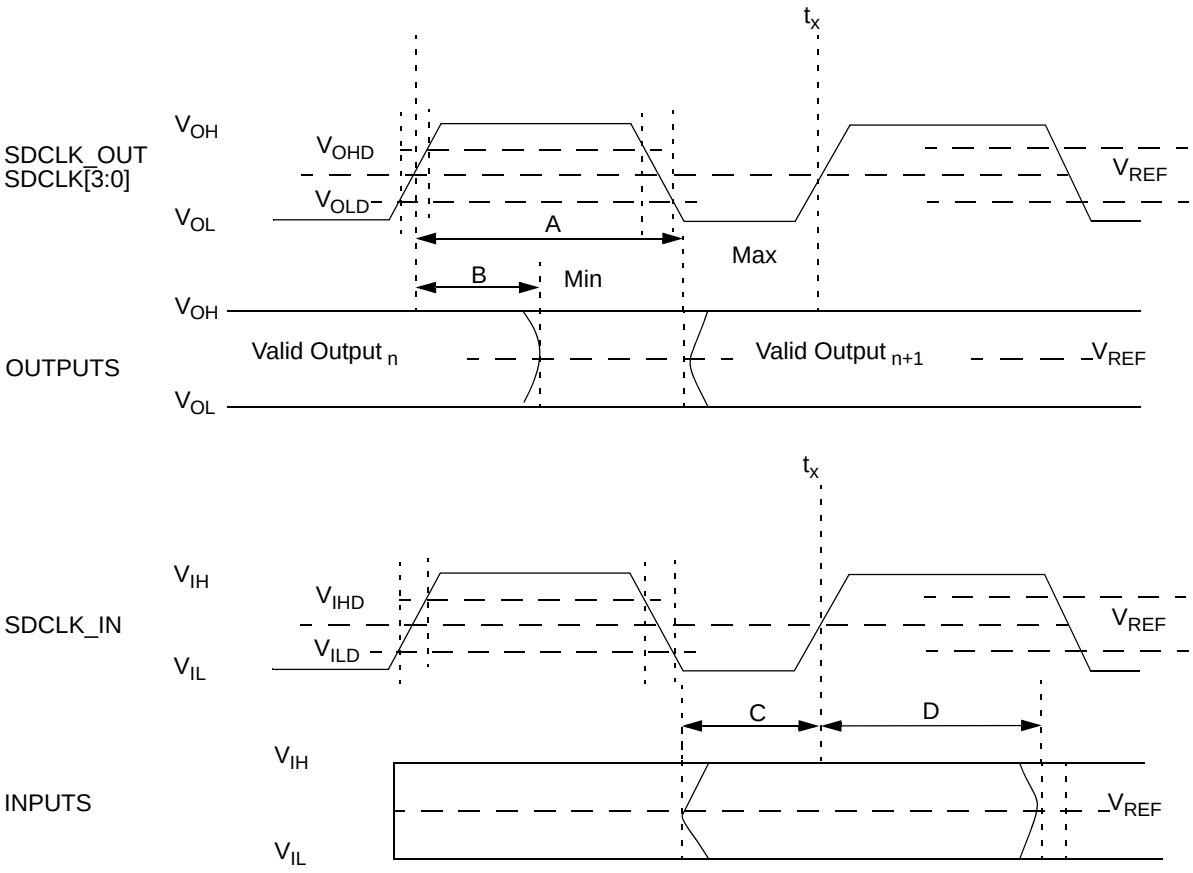
Legend: A = Maximum Output or Float Delay Specification
B = Minimum Output or Float Delay Specification
C = Minimum Input Setup Specification
D = Minimum Input Hold Specification

Figure 8-2. Drive level and Measurement Points for Switching Parameters

Electrical Specifications (Continued)

8.3.1 Memory Controller Interface

The minimum input setup and hold times described in Figure 8-3 (legend C and D) define the smallest acceptable sampling window during which a synchronous input signal must be stable to ensure correct operation.



Legend: A = Maximum Output Delay
B = Minimum Output Delay
C = Minimum Input Setup
D = Minimum Input Hold

Figure 8-3. Drive Level and Measurement Points for Switching Characters

Electrical Specifications (Continued)

Table 8-11. Memory Controller Timing Parameters

Symbol	Parameter	Min	Max	Unit
t_1	Control Output[1,2] Valid from SDCLK[3:0]	-3.0 + (x * y)	0.1 + (x * y)	ns
t_2	MA[12:0], BA[1.0] Output[2] Valid from SDCLK[3:0]	-3.2 + (x * y)	0.1 + (x * y)	ns
t_3	MD[63.0] Output[2] Valid from SDCLK[3:0]	-2.2 + (x * y)	0.7 + (x * y)	ns
t_4	MD[63.0] Read Data in Setup to SDCLK_IN	1.3		ns
t_5	MD[63:0] Read Data Hold to SDCLK_IN	2.0		ns
t_6	SDCLK[3:0], SDCLK_OUT cycle time			
	233 MHz	10	14	ns
	266 MHz	8.3	13.5	
	300 MHz	7.3	12.5	
t_7	SDCLK[3:0], SDCLK_OUT fall/rise time between (V_{OLD}-V_{OHD})		2	ns
t_9	SDCLK_IN fall/rise time between (V_{ILD}-V_{IHD})		2	ns
t_{10}	SDCLK[3:0], SDCLK_OUT high time			
	233 MHz	4.0		ns
	266 MHz	3.0		
	300 MHz	2.5		
t_{11}	SDCLK[3:0], SDCLK_OUT low time			
	233 MHz	4.0		ns
	266 MHz	2.5		
	300 MHz	2.5		

1. Control output includes all the following signals: RASA#, CASA#, WEA#, CKEA, DQM[7:0], and CS[1:0]#. Load = 50 pF, V_{CORE} = 1.8V@ 233/266 MHz, V_{CORE} = 2.1V@ 300 MHz, V_{IO} = 3.3V, @25°C.
2. Use the Min/Max equations `[value+(x * y)]` to calculate the actual value.
 x is the shift value which is applied to the SHFTSDCLK field, and y is 0.45 the core clock period.
 Note that the SHFTSDCLK field = GX_BASE+Memory Offset 8404h[5:3]. Refer to the *GX1 Processor Series Datasheet* for more information.
 For example, for a 266 MHz SC2200 running a 88.7 MHz SDRAM clock, with a shift value of 3:
 t1 Min = -3 + (3 * (3.76 * 0.45)) = 2.08 ns
 t1 Max = 0.1 + (3 * (3.76 * 0.45)) = 5.18 ns

Electrical Specifications (Continued)

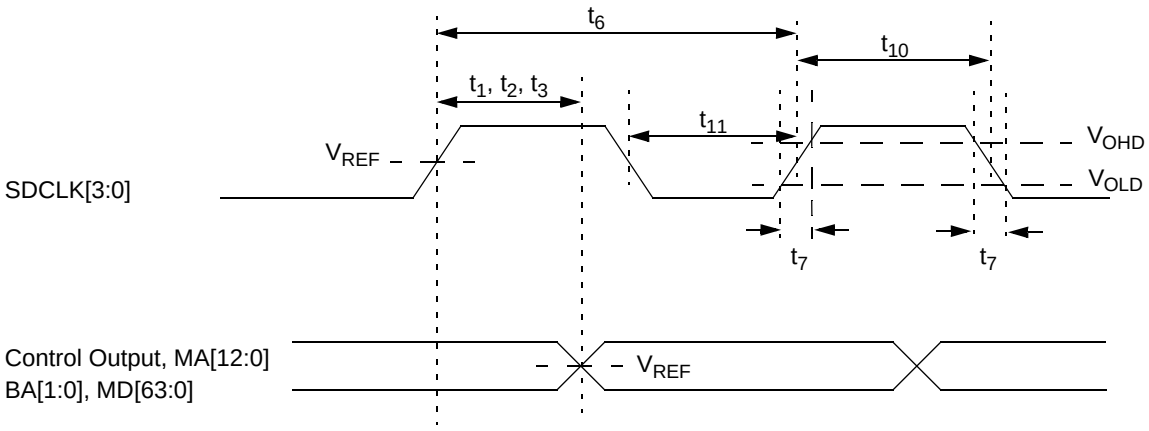


Figure 8-4. Memory Controller Output Valid Timing Diagram

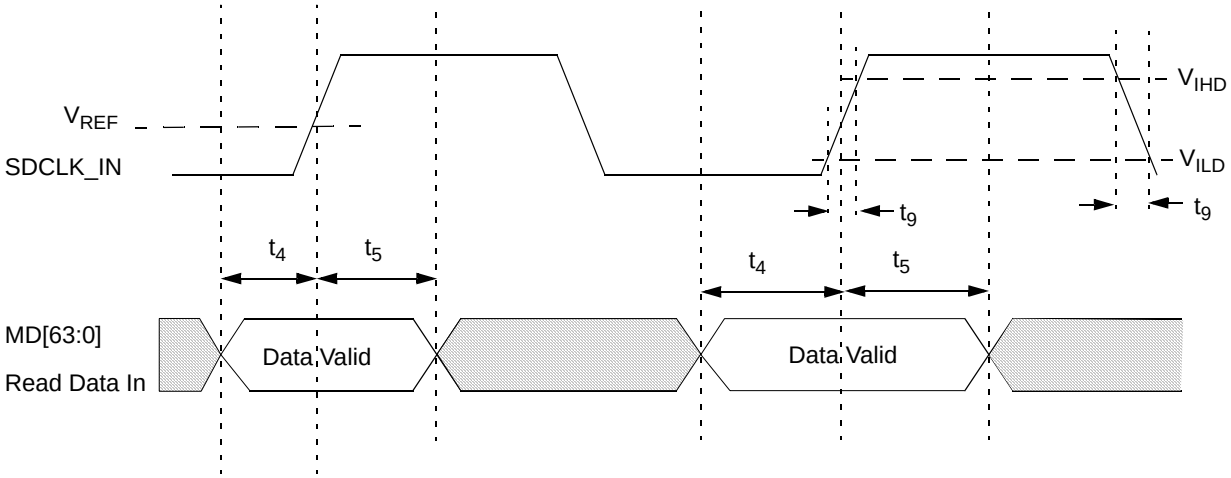


Figure 8-5. Read Data In Setup and Hold Timing Diagram

Electrical Specifications (Continued)

8.3.2 Video Port

Table 8-12. Video Input Port Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{VP_C}	VPCKIN cycle time	18		ns	
t_{VP_S}	Video Port input setup time before VPCKIN rising edge	6		ns	
t_{VP_H}	Video Port input hold time after VPCKIN Rising edge	0		ns	
t_{VPCK_FR}[1]	VPCKIN fall/rise time	-	2	ns	
t_{VPCK_D}	VPCKIN duty cycle	35/65		%	

1. Guaranteed by characterization.

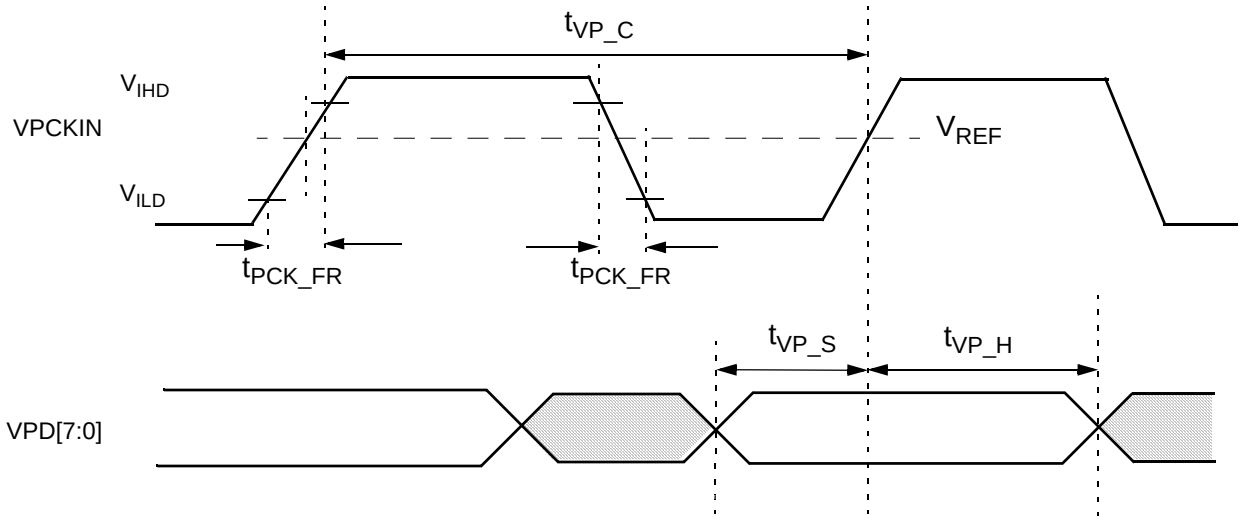


Figure 8-6. Video Input Port Timing Diagram

Electrical Specifications (Continued)

8.3.3 CRT and TFT Interface

Table 8-13 and Figure 8-7 describe the timing of the digital CRT interface of the SC2200. All measurement points in this table are identical to the voltage measurement levels described in Table 8-10 on page 370.

Note that signals DDC_SCL and DDC_SDA of the CRT interface are compliant with standard ACCESS.bus timing and are controlled by software.

Table 8-13. TFT Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{OV}	TFTD[17:0], TFTDE valid time after TFTDCK rising edge (multiplexed on IDE)	0	8	ns	
t_{OV}	TFTD[17:0], TFTDE valid time after TFTDCK rising edge (multiplexed on Parallel Port)	0	4	ns	
t_{CLK_RF}	TFTDCK rise/fall time[1] between 0.8V and 2.0V		3	ns	
t_{CLK_P}	TFTDCK period time (multiplexed on IDE)	25		ns	
t_{CLK_P}	TFTDCK period time (muxed on Parallel Port)	12.5		ns	
t_{CLK_D}	TFTDCK duty cycle	40/60		%	

1. Guaranteed by characterization

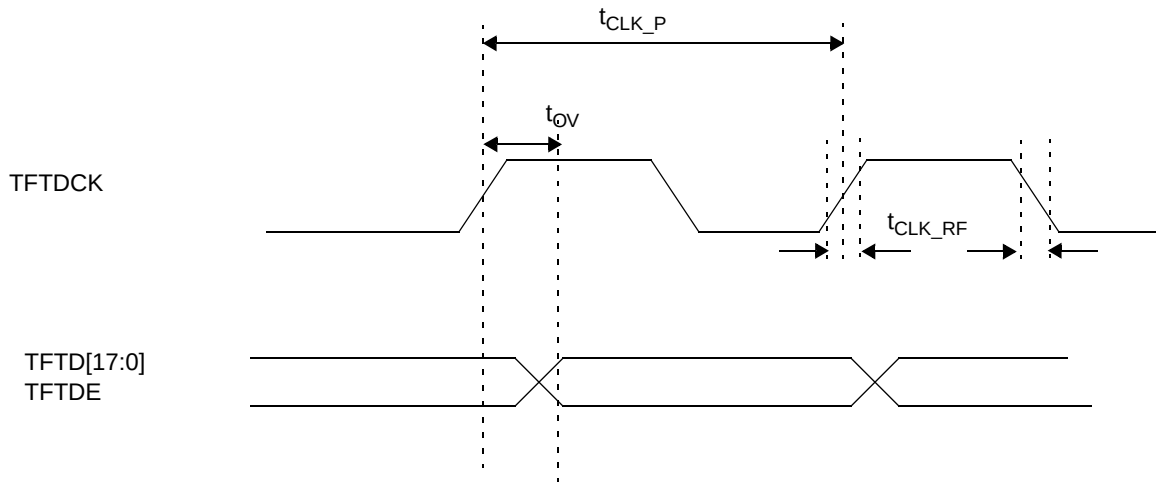


Figure 8-7. TFT Timing Diagram

Electrical Specifications (Continued)

Table 8-14. CRT VESA Compatible[1] DAC (RED, GREEN, and BLUE Outputs)

Symbol	Parameter	Min	Max	Unit	Comments
V_{FR}	Full range output voltage	0.6	0.72	V	SETRES = 470 R_L = 37.5 Digital input = FFh
I_{FR}	Full range output current	16	19.2	mA	SETRES = 470 R_L = 37.5 Digital input = FFh
INL	Integral linearity error[2]		±1	LSB	
DNL	Differential linearity error[3]		±1	LSB	
t_{ST}	Full-scale settling time[4]		10	ns	C_L = 40 pF
t_R	Rise time[5]		4	ns	
DDM	DAC to DAC matching		5	%	
C_{OUT}	Max output capacitance		15	pF	
PSRR	Power supply rejection ratio[6]		3.5	%	At 0 to 1 MHz

1. Black level = Blank level = 0 mA, 0V.
2. The maximum difference between the ideal (straight) conversion line and the actual conversion curve.
3. The maximum difference between the ideal step size (1 LSB) and any actual step size.
4. The input changes from 00h to FFh. The time from output voltage at 50% of step change to output settling (within an error of ±1 LSB) is the full-scale settling time.
5. The input changes from 00h to FFh. The output changes from 10% to 90%.
6. AV_{CCRT} changes within the range of 3V to 3.6V. Output voltage is measured for peak-to-peak maximum change. PSSR is the ratio of the measurement of output at AV_{CCRT} = 3.3V.

Electrical Specifications (Continued)

8.3.4 ACCESS.bus Interface

The following tables describe the timing for all ACCESS.bus signals.

Notes: 1) All ACCESS.bus timing is not 100% tested.

2) In this table t_{CLK} = 1/24 MHz = 41.7 ns.

Table 8-15. ACCESS.bus Input Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{BUFi}	Bus free time between Stop and Start condition	$t_{SCLhigho}$			
t_{CSTOsi}	AB1C/AB2C setup time	$8 * t_{CLK} - t_{SCLri}$			Before Stop condition
t_{CSTRhi}	AB1C/AB2C hold time	$8 * t_{CLK} - t_{SCLri}$			After Start condition
t_{CSTRsi}	AB1C/AB2C setup time	$8 * t_{CLK} - t_{SCLri}$			Before Start condition
t_{DHCsi}	Data high setup time	$2 * t_{CLK}$			Before AB1C/AB2C rising edge
t_{DLCsi}	Data low setup time	$2 * t_{CLK}$			Before AB1C/AB2C rising edge
t_{SCLfi}	AB1D/AB2D fall time		300	ns	
t_{SCLri}	AB1D/AB2D rise time		1	µs	
$t_{SCLlowi}$	AB1C/AB2C low time	$16 * t_{CLK}$			After AB1C/AB2C falling edge
$t_{SCLhighi}$	AB1C/AB2C high time	$16 * t_{CLK}$			After AB1C/AB2C rising edge
t_{SDAfi}	AB1D/AB2D fall time		300	ns	
t_{SDAri}	AB1D/AB2D rise time		1	µs	
t_{SDAhi}	AB1D/AB2D hold time	0			After AB1C/AB2C falling edge
t_{SDAsi}	AB1D/AB2D setup time	$2 * t_{CLK}$			Before AB1C/AB2C rising edge

Table 8-16. ACCESS.bus Output Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
$t_{SCLhigho}$	AB1C/AB2C high time	$K * t_{CLK} - 1$ µs			After AB1C/AB2C rising edge[1]
$t_{SCLlowo}$	AB1C/AB2C low time	$K * t_{CLK} - 1$ µs			After AB1C/AB2C falling edge
t_{BUFo}	Bus free time between Stop and Start condition	$t_{SCLhigho}$[2]	1	µs	
t_{CSTOso}	AB1C/AB2C setup time	$t_{SCLhigho}$[2]	1	µs	Before Stop condition
t_{CSTRho}	AB1C/AB2C hold time	$t_{SCLhigho}$[2]	1	µs	After Start condition
t_{CSTRso}	AB1C/AB2C setup time	$t_{SCLhigho}$[2]	1	µs	Before Start condition
t_{DHCso}	Data high setup time	$t_{SCLhigho}$[2] - t_{SDAro}	1	µs	Before AB1C/AB2C rising edge
t_{DLCso}	Data low setup time	$t_{SCLhigho}$[2] - t_{SDAfo}	1	µs	Before AB1C/AB2C rising edge
t_{SCLfo}	AB1D/AB2D signal fall time		300	ns	
t_{SCLro}	AB1D/AB2D signal rise time		1	µs	

Electrical Specifications (Continued)

Table 8-16. ACCESS.bus Output Timing Parameters (Continued)

Symbol	Parameter	Min	Max	Unit	Comments
t_{SDAfo}	AB1D/AB2D signal fall time		300	ns	
t_{SDAro}	AB1D/AB2D signal rise time		1	µs	
t_{SDAho}	AB1D/AB2D hold time	$7 * t_{CLK} - t_{SCLfo}$			After AB1C/AB2C falling edge
t_{SDAvo}	AB1D/AB2D valid time		$7 * t_{CLK} + t_{RD}$		After AB1C/AB2C falling edge

1. K is determined by bits [7:1] of the ACBCTL2 register (LDN 05h/06h, Offset 05h).
2. This value depends on the signal capacitance and the pull-up value of the relevant pin.

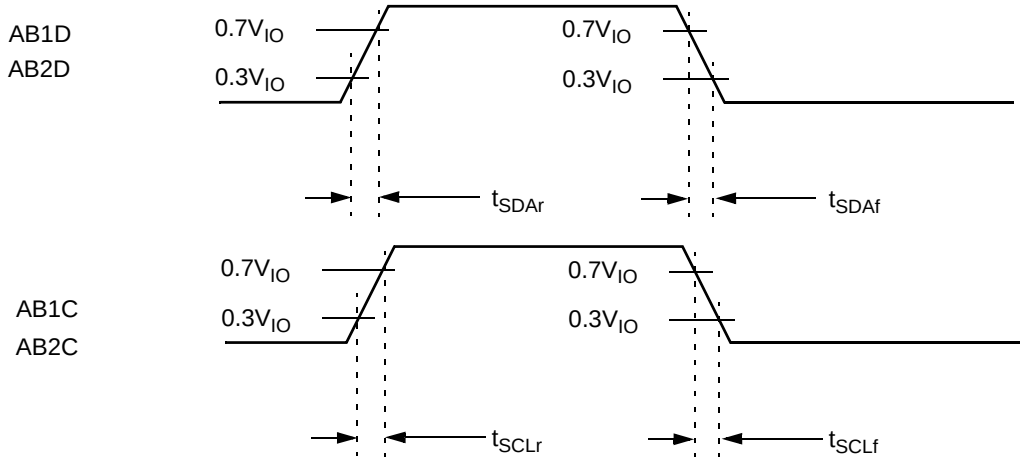


Figure 8-8. ACB Signals: Rising Time and Falling Timing Diagram

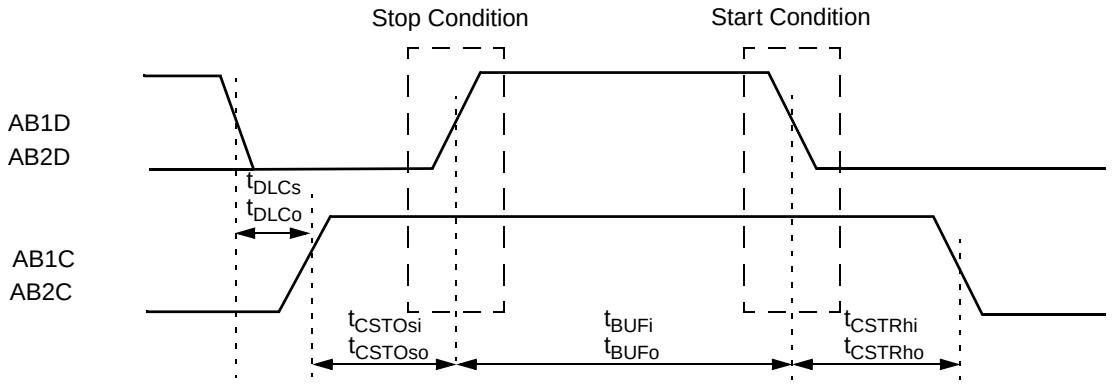


Figure 8-9. ACB Start and Stop Condition Timing Diagram

Electrical Specifications (Continued)

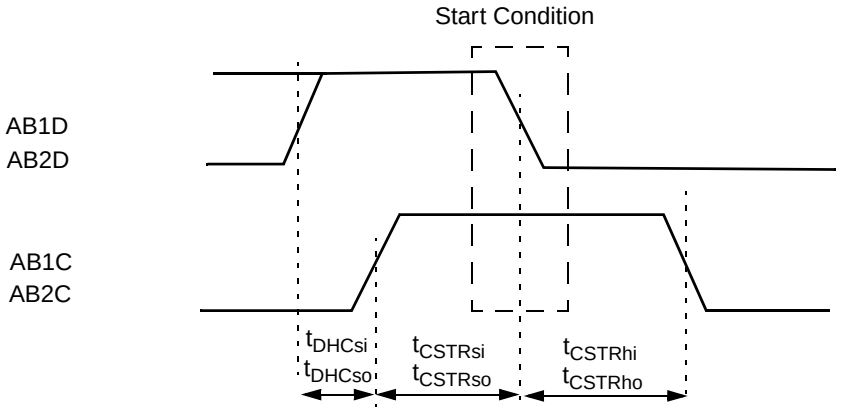


Figure 8-10. ACB Start Condition Timing Diagram

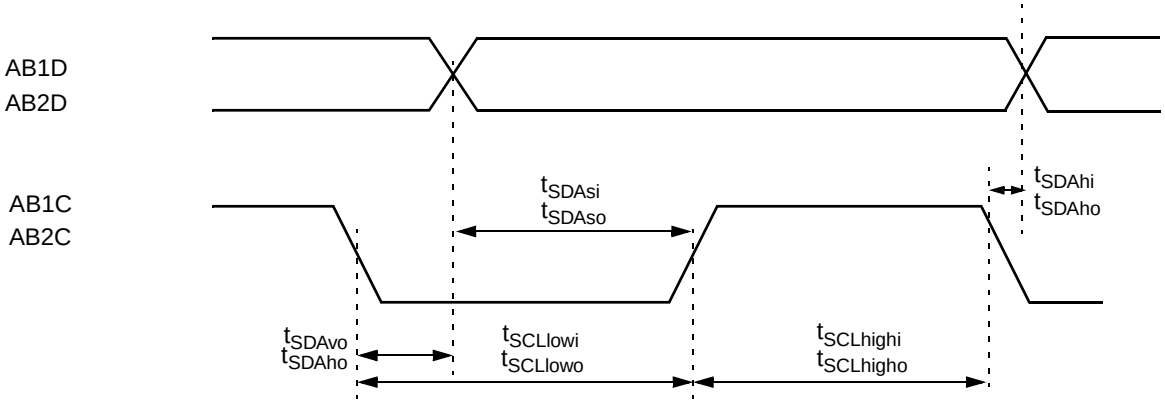


Figure 8-11. ACB Data Bit Timing Diagram

Electrical Specifications (Continued)

8.3.5 PCI Bus Interface

The SC2200 is compliant with PCI Bus Rev. 2.1 specifications. Relevant information from the PCI Bus specifications is provided below.

All parameters in Table 8-17 are not 100% tested. The parameters in this table are further described in Figure 8-13.

Table 8-17. PCI AC Specifications

Symbol	Parameter	Min	Max	Unit	Comments
$I_{OH}(AC)$[1, 2]	Switching Current High	$-12V_{IO}$		mA	$0 < V_{OUT} \leq 0.3V_{IO}$
		$-17.1(V_{IO}-V_{OUT})$		mA	$0.3V_{IO} < V_{OUT} < 0.9V_{IO}$
			Equation A (Figure 8-13)		$0.7V_{IO} < V_{OUT} < V_{IO}$
	Test Point[2]		$-32V_{IO}$	mA	$V_{OUT} = 0.7V_{IO}$
$I_{OL}(AC)$[1]	Switching Current Low	$16V_{IO}$		mA	$V_{IO} > V_{OUT} \geq 0.6V_{IO}$
		$26.7V_{OUT}$		mA	$0.6V_{IO} > V_{OUT} > 0.1V_{IO}$[1]
			Equation B (Figure 8-13)		$0.18V_{IO} > V_{OUT} > 0$[1, 2]
	Test Point[2]		$38V_{IO}$	mA	$V_{OUT} = 0.18V_{IO}$
I_{CL}	Low Clamp Current	$-25+(V_{IN}+1)/0.015$		mA	$-3 < V_{IN} \leq -1$
I_{CH}	High Clamp Current	$25+(V_{IN}-V_{IO}-1)/0.015$		mA	$V_{IO}+4 > V_{IN} > V_{IO}+1$
$SLEW_R$[3]	Output Rise Slew Rate	1	4	V/ns	$0.2V_{IO}$ - $0.6V_{IO}$ Load
$SLEW_F$[3]	Output Fall Slew Rate	1	4	V/ns	$0.6V_{IO}$ - $0.2V_{IO}$ Load

1. Refer to the V/I curves in Figure 8-13. This specification does not apply to PCICLK0, PCICLK1, and PCIRST# which are system outputs.
2. Maximum current requirements are met when drivers pull beyond the first step voltage. Equations which define these maximum values (A and B) are provided with relevant diagrams in Figure 8-13. These maximum values are guaranteed by design.
3. Rise slew rate does not apply to open-drain outputs. This parameter is interpreted as the cumulative edge rate across the specified range, according to the test circuit in Figure 8-12.

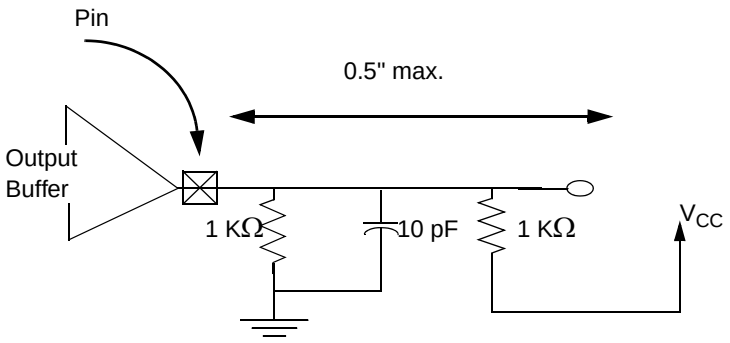
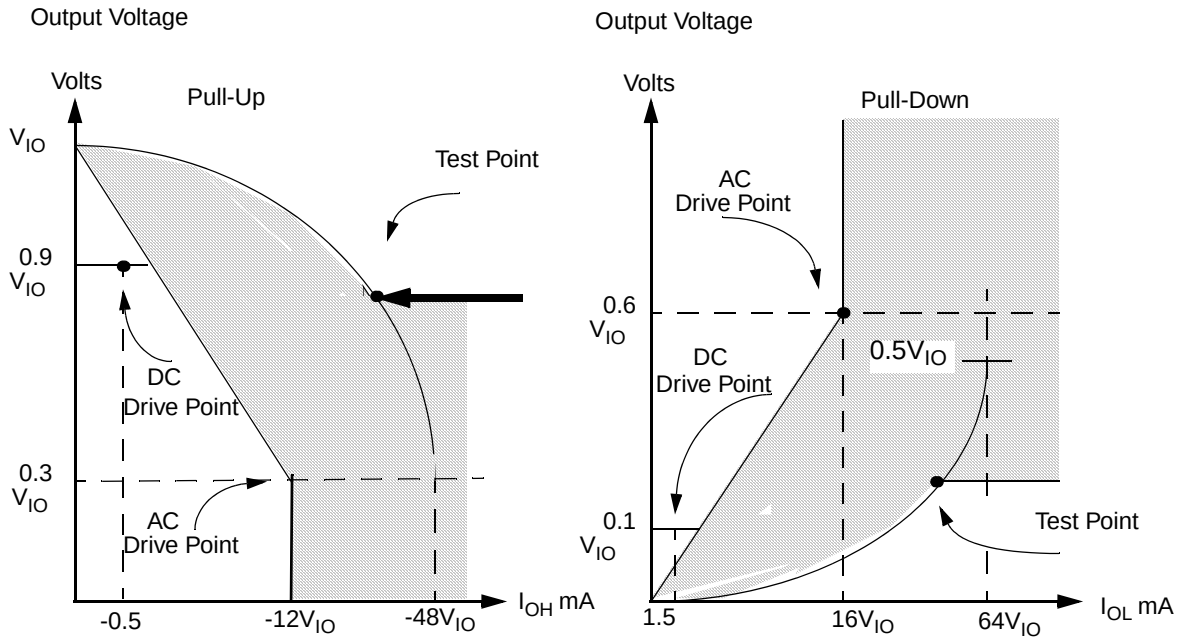


Figure 8-12. Testing Setup for Slew Rate and Minimum Timing

Electrical Specifications (Continued)



Equation A

$I_{OH} = (98.0/V_{IO})*(V_{OUT}-V_{IO})*(V_{OUT}+0.4V_{IO})$
for $V_{IO}>V_{OUT}>0.7V_{IO}$

Equation B

$I_{OL} = (256/V_{IO})*V_{OUT}*(V_{IO}-V_{OUT})$
for $0V<V_{OUT}<0.18V_{IO}$

Figure 8-13. V/I Curves for PCI Output Signals

Table 8-18. PCI Clock Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{CYC}	PCICLK Cycle time[1]	30		ns	
t_{HIGH}	PCICLK High time[2]	11		ns	
t_{LOW}	PCICLK Low time[2]	11		ns	
$PCICLK_{sr}$	PCICLK Slew Rate[3]	1	4	V/ns	
$PCIRST_{sr}$	PCIRST# Slew Rate[4]	50	-	mV/ns	

1. Clock frequency is between nominal DC and 33 MHz. Device operational parameters at frequencies under 16 MHz are not 100% tested. The clock can only be stopped in a low state.
2. Guaranteed by characterization.
3. Slew rate must be met across the minimum peak-to-peak portion of the clock waveform (see Figure 8-14).
4. The minimum PCIRST# slew rate applies only to the rising (de-assertion) edge of the reset signal. See Figure 8-18 for PCIRST# timing.

Electrical Specifications (Continued)

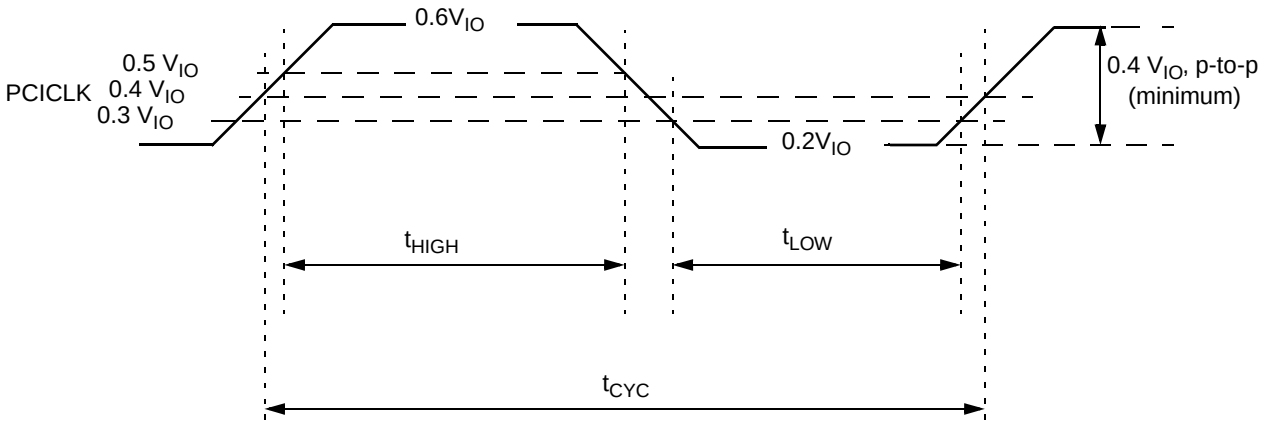


Figure 8-14. PCICLK Timing and Measurement Points

Table 8-19. PCI Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{VAL}	PCICLK to Signal Valid Delay[1,2] (on the bus)	2	11	ns	
$t_{VAL}(ptp)$	PCICLK to Signal Valid Delay[1,2] (GNT#)	2	9	ns	
t_{ON}	Float to Active Delay[1,3]	2		ns	
t_{OFF}	Active to Float Delay[1,3]		28	ns	
t_{SU}	Input Setup Time to PCICLK[4] (on the bus)	7		ns	
$t_{SU}(ptp)$	Input Setup Time to PCICLK[4] (REQ#)	6		ns	
t_{H}	Input Hold Time from PCICLK[4]	0		ns	
t_{RST}	PCIRST# Active Time After Power Stable[5,3]	1		ms	
$t_{RST-CLK}$	PCIRST# Active Time After PCICLK Stable[5,3]	100		µs	
$t_{RST-OFF}$	PCIRST# Active to Output Float Delay[3,5,6]		40	ns	

1. See the timing measurement conditions in Figure 8-16.
2. Minimum times are evaluated with same load used for slew rate measurement (as shown in note 3 of Table); maximum times are evaluated with the load circuits shown in Figure 8-15, for high-going and low-going edges respectively.
3. Not 100% tested.
4. See the timing measurement conditions in Figure 8-17.
5. PCIRST# is asserted and de-asserted asynchronously with respect to PCICLK (see Figure 8-18).
6. All output drivers are asynchronously floated when PCIRST# is active.

Electrical Specifications (Continued)

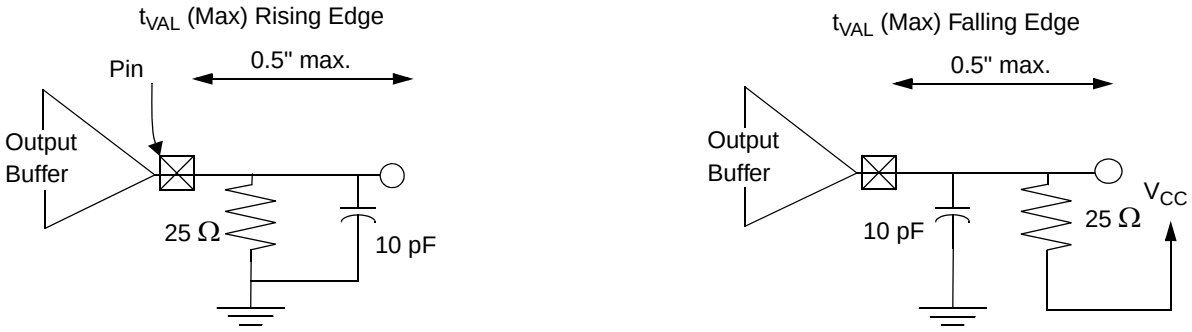


Figure 8-15. Load Circuits for Maximum Time Measurements

8.3.5.1 Measurement and Test Conditions

Table 8-20. Measurement Condition Parameters

Symbol	Value	Unit
V_{TH}[1]	0.6 V_{IO}	V
V_{TL}[1]	0.2 V_{IO}	V
V_{TEST}	0.4 V_{IO}	V
V_{STEP} (Rising Edge)	0.285 V_{IO}	V
V_{STEP} (Falling Edge)	0.615 V_{IO}	V
V_{MAX}[2]	0.4 V_{IO}	V
Input Signal Edge Rate	1	V/ns

1. The input test is performed with 0.1 V_{IO} of overdrive. Timing parameters must not exceed this overdrive.
2. V_{MAX} specifies the maximum peak-to-peak waveform allowed for measuring input timing.

Electrical Specifications (Continued)

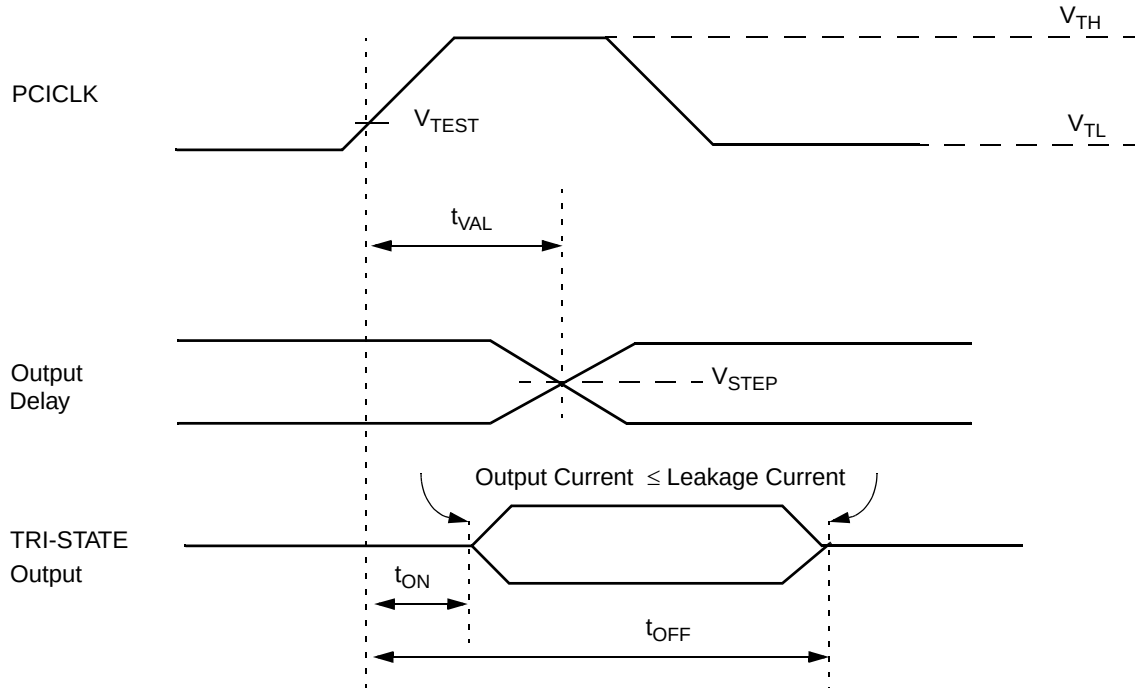


Figure 8-16. Output Timing Measurement Conditions

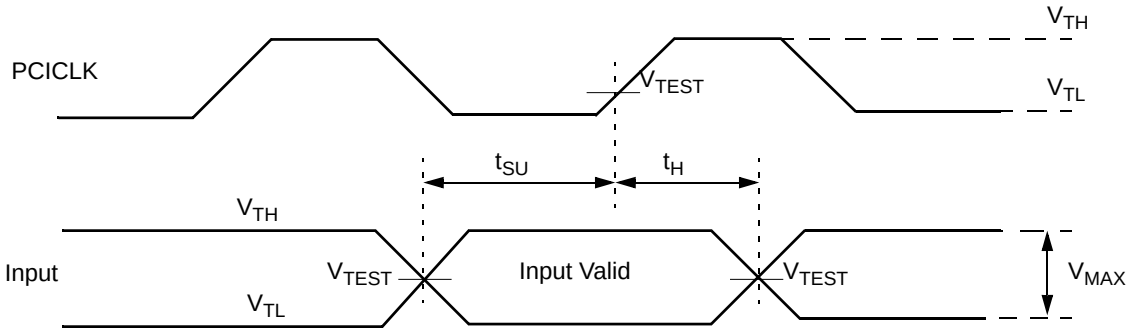
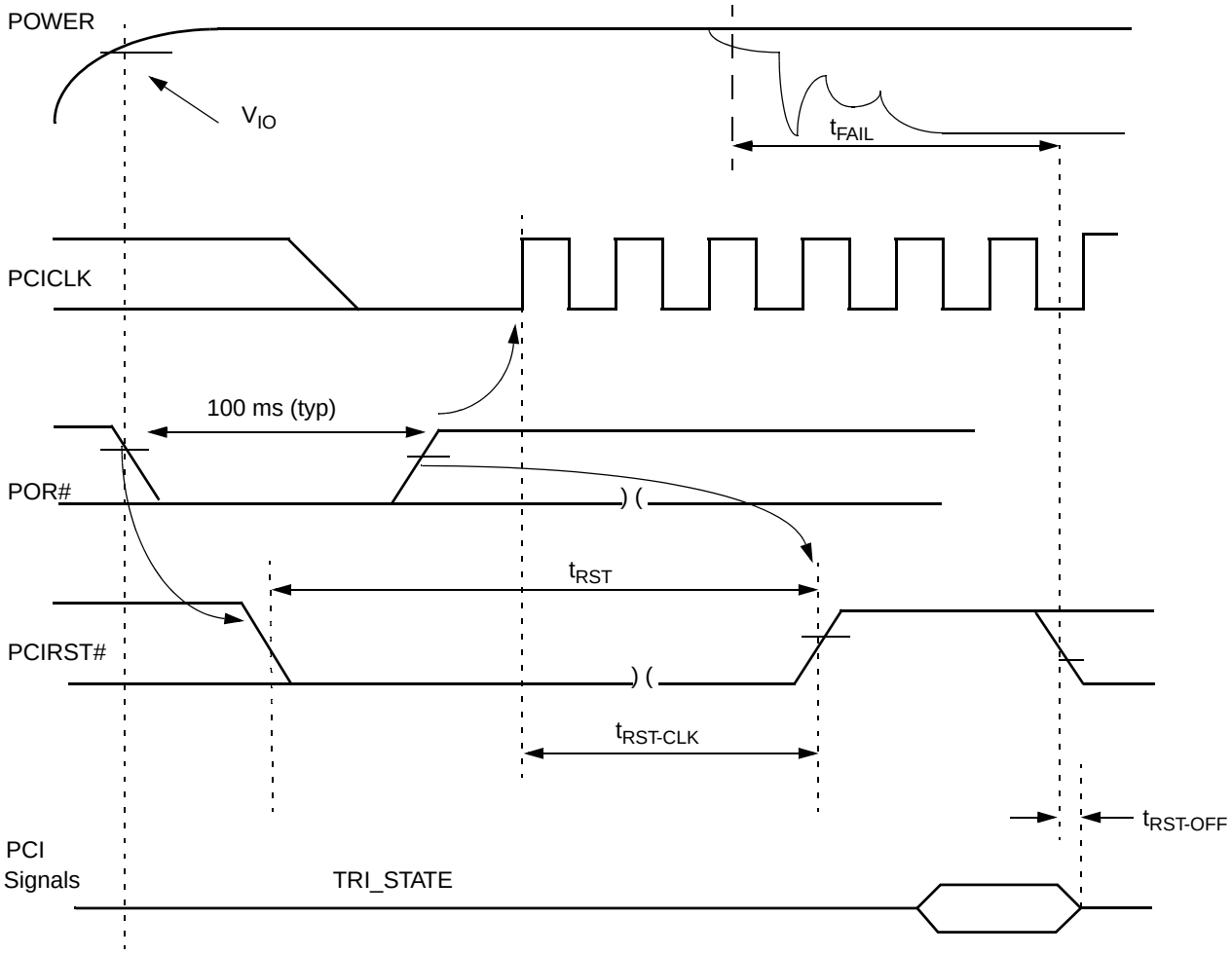


Figure 8-17. Input Timing Measurement Conditions

Electrical Specifications (Continued)



Note: The value of t_{FAIL} is 500 ns (maximum) from the power rail which exceeds specified tolerance by more than 500 mV.

Figure 8-18. PCI Reset Timing

Electrical Specifications (Continued)

8.3.6 Sub-ISA Interface

All output timing is guaranteed for 50 pF load, unless otherwise specified.

The ISA Clock divisor (defined in F0 Index 50h[2:0] of the Core Logic module) is 011.

Table 8-21. Sub-ISA Timing Parameters

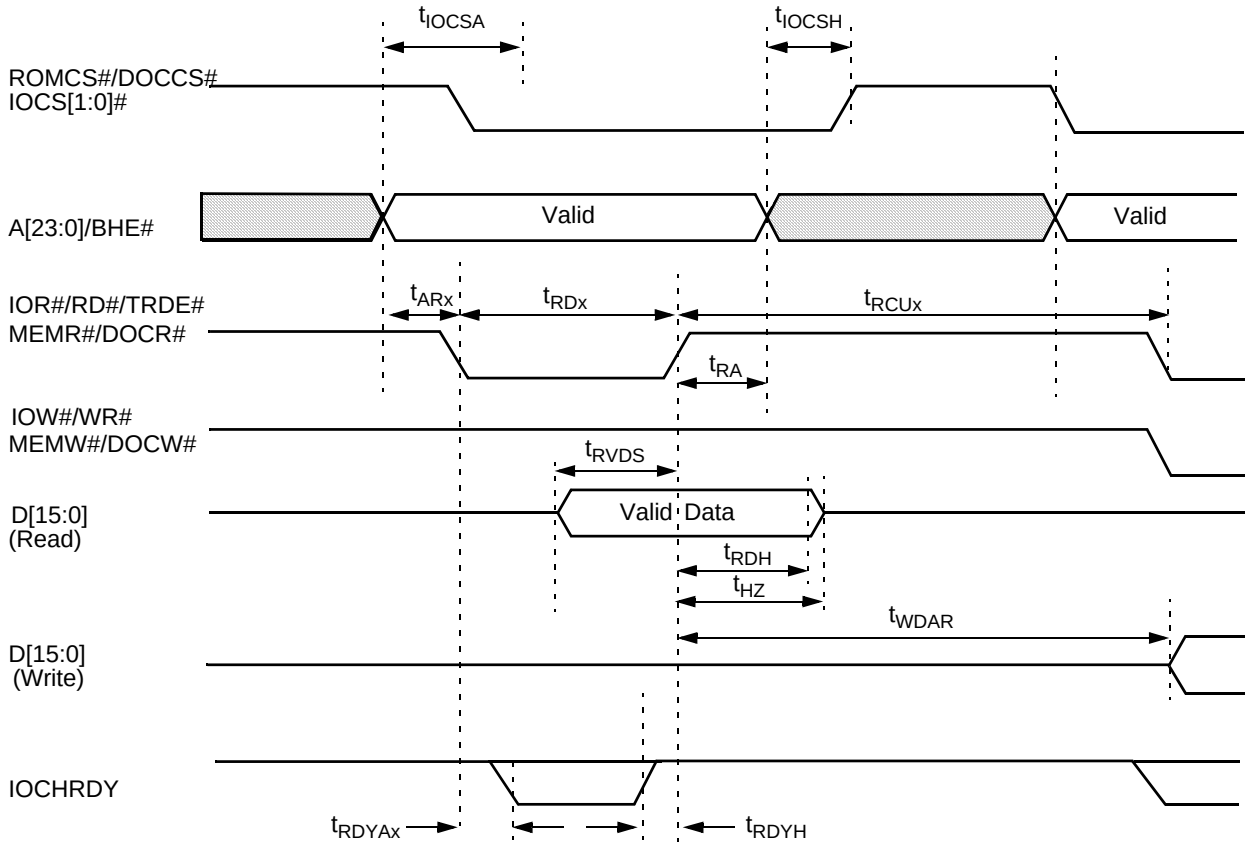
Symbol	Parameter	Bus Width (Bits)	Type	Min (ns)	Max (ns)	Figure	Comments
t_{RD1}	MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	16	M	225		8-19	Standard
t_{RD2}	MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	16	M	105		8-19	Zero wait state
t_{RD3}	IOR#/RD#/TRDE# Read active pulse width FE to RE	16	I/O	160		8-19	Standard
t_{RD4}	IOR#/MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	8	M, I/O	520		8-19	Standard
t_{RD5}	IOR#/MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	8	M, I/O	160		8-19	Zero wait state
t_{RCU1}	MEMR#/DOCR#/RD#/TRDE# inactive pulse width	16	M	103		8-19	
t_{RCU2}	MEMR#/DOCR#/RD#/TRDE# inactive pulse width	8	M	163		8-19	
t_{RCU3}	IOR#/RD#/TRDE# inactive pulse width	8, 16	I/O	163		8-19	
t_{WR1}	MEMW#/WR# Write active pulse width FE to RE	16	M	225		8-20	Standard
t_{WR2}	MEMW#/DOCW#/WR# Write active pulse width FE to RE	16	M	105		8-20	Zero wait state
t_{WR3}	IOW#/WR# Write active pulse width FE to RE	16	I/O	160		8-20	Standard
t_{WR4}	IOW#/MEMW#/DOCW#/WR# Write active pulse width FE to RE	8	M, I/O	520		8-20	Standard
t_{WR5}	IOW#/MEMW#/DOCW#/WR# Write active pulse width FE to RE	8	M, I/O	160		8-20	Zero wait state
t_{WCU1}	MEMW#/WR#/DOCW# inactive pulse width	16	M	103		8-20	
t_{WCU2}	MEMW#/WR#/DOCW# inactive pulse width	8	M	163		8-20	
t_{WCU3}	IOW#/WR# inactive pulse width	8, 16	I/O	163		8-20	
t_{RDYH}	IOR#/MEMR#/RD#/DOCR#/IOW#/ MEMW#/WR#/DOCW# Hold after IOCHRDY RE	8, 16	M, I/O	120		8-19 8-20	
t_{RDYA1}	IOCHRDY valid after IOR#/MEMR#/ RD#/DOCR#/IOW#/MEMW#/WR#/ DOCW# FE	16	M, I/O		78	8-19 8-20	

Electrical Specifications (Continued)

Table 8-21. Sub-ISA Timing Parameters (Continued)

Symbol	Parameter	Bus Width (Bits)	Type	Min (ns)	Max (ns)	Figure	Comments
t_{RDYA2}	IOCHRDY valid after IOR#/MEMR#/ RD#/DOCR#/IOW#/MEMW#/WR#/ DOCW# FE	8	M, I/O		366	8-19 8-20	
t_{IOCSA}	IOCS[1:0]#/DOCS#/ROMCS# driven active from A[23:0] valid	8, 16	M, I/O		34	8-19 8-20	
t_{IOCSH}	IOCS[1:0]#/DOCS#/ROMCS# valid Hold after A[23:0] invalid	8, 16	M, I/O	0		8-19 8-20	
t_{AR1}	A[23:0]/BHE# valid before MEMR#/DOCR# active	16	M	34		8-19	
t_{AR2}	A[23:0]/BHE# valid before IOR# active	16	I/O	100		8-19	
t_{AR3}	A[23:0]/BHE# valid before MEMR#/DOCR#/IOR# active	8	M, I/O	100		8-19	
t_{RA}	A[23:0]/BHE# valid Hold after MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O	25		8-19	
t_{RVDS}	Read data D[15:0] valid setup before MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O	24		8-19	
t_{RDH}	Read data D[15:0] valid Hold after MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O	0		8-19	
t_{HZ}	Read data floating after MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O		41	8-19	
t_{AW1}	A[23:0]/BHE# valid before MEMW#/DOCW# active	16	M	34		8-20	
t_{AW2}	A[23:0]/BHE# valid before IOW# active	16	I/O	100		8-20	
t_{AW3}	A[23:0]/BHE# valid before MEMW#/DOCW#/IOW# active	8	M, I/O	100		8-20	
t_{WA}	A[23:0]/BHE# valid Hold after MEMW#/DOCW#/IOW# invalid	8, 16	M, I/O	25		8-20	
t_{DV1}	Write data D[15:0] valid after MEMW#/DOCW# active	8, 16	M	40		8-20	
t_{DV2}	Write data D[15:0] valid after IOW# active	8	I/O	40		8-20	
t_{DV3}	Write data D[15:0] valid after IOW# active	16	I/O	-23		8-20	
t_{WTR}	TRDE# inactive after MEMW#/DOCW#/IOW# inactive	8, 16	M, I/O	20		8-20	
t_{DH}	Write data D[15:0] after MEMW#/DOCW#/IOW# inactive	8, 16	M, I/O	45		8-20	
t_{DF}	Write data D[15:0] goes TRI-STATE after MEMW#/DOCW#/IOW# inactive	8, 16	M, I/O		105	8-20	
t_{WDAR}	Write data D[15:0] after read MEMR#/DOCR#/IOR#	8, 16	M, I/O	41		8-19	

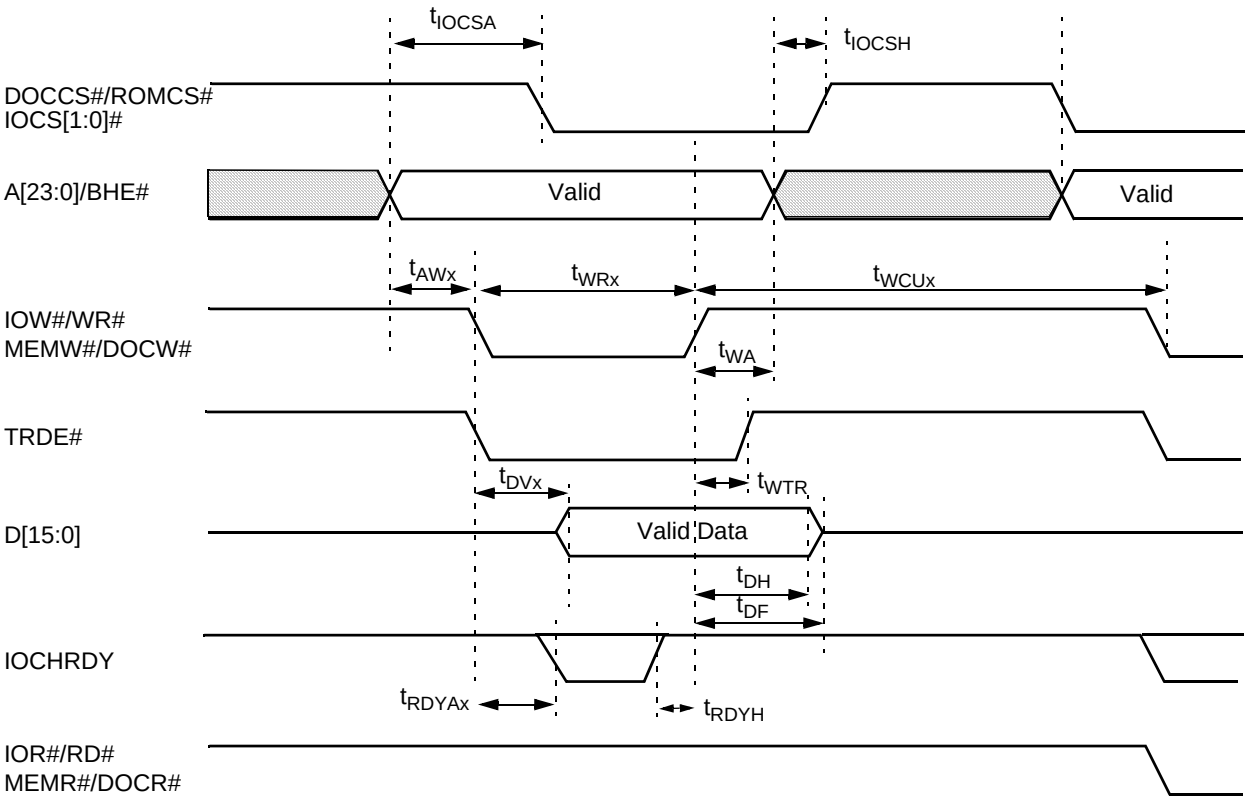
Electrical Specifications (Continued)



Note: x indicates a numeric index for the relevant symbol.

Figure 8-19. Sub-ISA Read Operation Timing Diagram

Electrical Specifications (Continued)



Note: x indicates a numeric index for the relevant symbol.

Figure 8-20. Sub-ISA Write Operation Timing Diagram

Electrical Specifications (Continued)

8.3.7 LPC Interface

Table 8-22. LPC and SERIRQ Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{VAL}	Output Valid delay	0	17	ns	After PCICLK rising edge
t_{ON}	Float to Active delay	2		ns	After PCICLK rising edge
t_{OFF}	Active to Float delay		28	ns	After PCICLK rising edge
t_{SU}	Input Setup time	7		ns	Before PCICLK rising edge
t_{HI}	Input Hold time	0		ns	After PCICLK rising edge

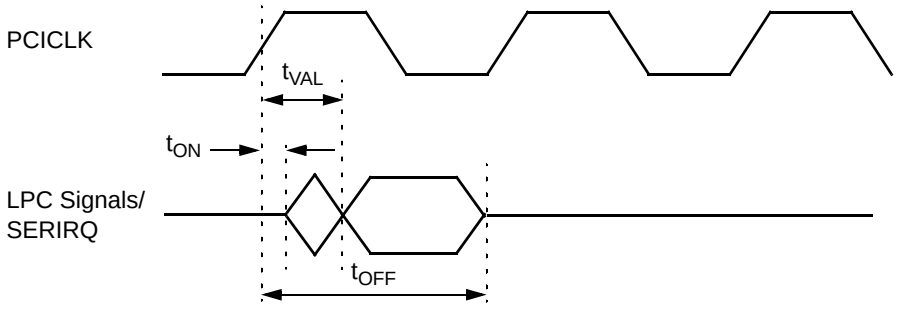


Figure 8-21. LPC Output Timing Diagram

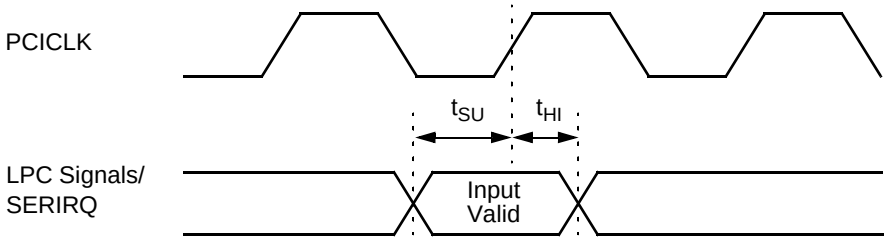


Figure 8-22. LPC Input Timing Diagram

Electrical Specifications (Continued)

8.3.8 IDE Interface

Table 8-23. IDE General Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{IDE_FALL}	Fall time of all IDE signals. From $0.9V_{IO}$ to $0.1V_{IO}$	5		ns	C_L = 40 pF
t_{IDE_RISE}	Rise time of all IDE signals. From $0.1V_{IO}$ to $0.9V_{IO}$	5		ns	C_L = 40 pF
$t_{IDE_RST_PW}$	IDE_RST# pulse width	25		µs	

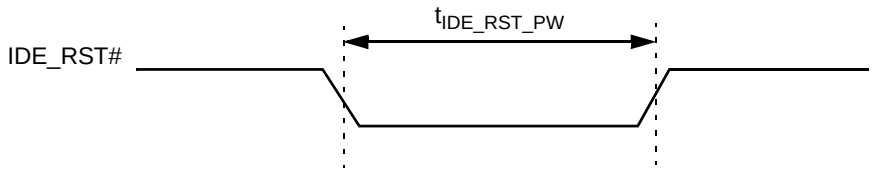


Figure 8-23. IDE Reset Timing Diagram

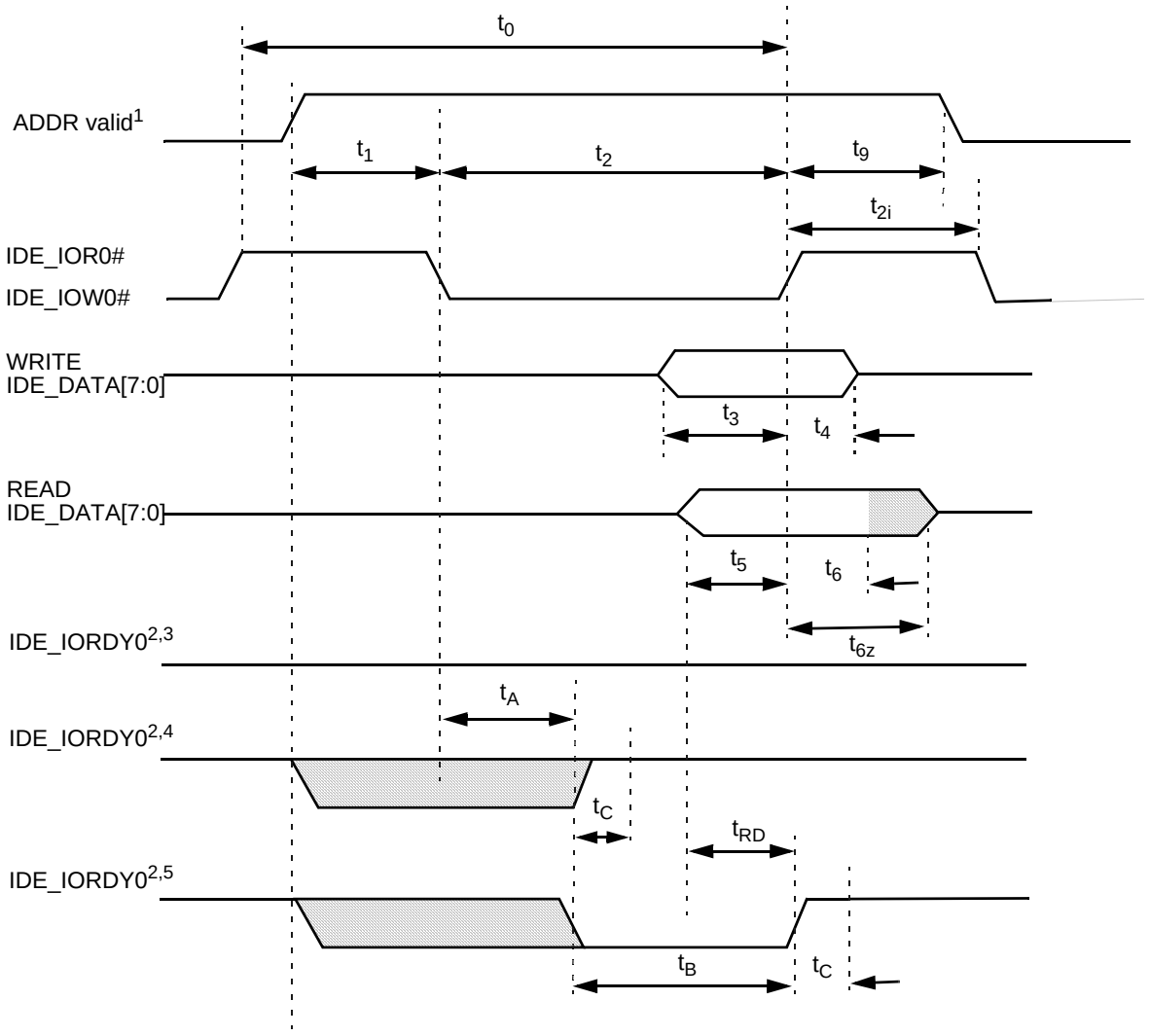
Electrical Specifications (Continued)

Table 8-24. Register Transfer to/from Device Timing Parameters

Symbol	Parameter	Mode 0 (ns)	Mode 1 (ns)	Mode 2 (ns)	Mode 3 (ns)	Mode 4 (ns)
t_0	Cycle time[1] (min)	600	383	240	180	120
t_1	Address valid to IDE_IOR[0:1]#/ IDE_IOW[0:1]# setup (min)	70	50	30	30	25
t_2	IDE_IOR[0:1]#/IDE_IOW[0:1]# pulse width 8-bit[1] (min)	290	290	290	80	70
t_{2i}	IDE_IOR[0:1]#/IDE_IOW[0:1]# recovery time[1] (min)	-	-	-	70	25
t_3	IDE_IOW[0:1]# data setup (min)	60	45	30	30	20
t_4	IDE_IOW[0:1]# data hold (min)	30	20	15	10	10
t_5	IDE_IOR[0:1]# data setup (min)	50	35	20	20	20
t_6	IDE_IOR[0:1]# data hold (min)	5	5	5	5	5
t_{6Z}	IDE_IOR[0:1]# data TRI-STATE[2] (max)	30	30	30	30	30
t_9	IDE_IOR[0:1]#/IDE_IOW[0:1]# to address valid hold (min)	20	15	10	10	10
t_{RD}	Read Data Valid to IDE_IORDY[0:1] active (if IDE_IORDY[0:1] initially low after t_A (min)	0	0	0	0	0
t_A	IDE_IORDY[0:1] Setup time[3]	35	35	35	35	35
t_B	IDE_IORDY[0:1] Pulse Width (max)	1250	1250	1250	1250	1250
t_C	IDE_IORDY[0:1] assertion to release (max)	5	5	5	5	5

1. t_0 is the minimum total cycle time, t_2 is the minimum command active time, and t_{2i} is the minimum command recovery time or command inactive time. The actual cycle time equals the sum of the command active time and the command inactive time. The three timing requirements of t_0, t_2, and t_{2i} are met. The minimum total cycle time requirements is greater than the sum of t_2 and t_{2i}. (This means that a host implementation can lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data.)
2. This parameter specifies the time from the rising edge of IDE_IOR[0:1]# to the time that the data bus is no longer driven by the device (TRI-STATE).
3. The delay from the activation of IDE_IOR[0:1]# or IDE_IOW[0:1]# until the state of IDE_IORDY[0,1] is first sampled. If IDE_IORDY[0:1] is inactive, then the host waits until IDE_IORDY[0:1] is active before the PIO cycle is completed. If the device is not driving IDE_IORDY[0:1] negated after activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_5 is met and t_{RD} is not applicable. If the device is driving IDE_IORDY[0:1] negated after activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_{RD} is met and t_5 is not applicable.

Electrical Specifications (Continued)



Notes:

1) Device address consists of signals IDE_CS[0:1]# and IDE_ADDR[2:0].

2) Negation of IDE_IORDY0,1 is used to extend the PIO cycle. The determination of whether or not the cycle is to be extended is made by the host after t_A from the assertion of IDE_IOR[0:1]# or IDE_IOW[0:1]#.

3) Device never negates IDE_IORDY[0:1]. Device keeps IDE_IORDY[0:1] released, and no wait is generated.

4) Device negates IDE_IORDY[0:1] before t_A but causes IDE_IORDY[0:1] to be asserted before t_A. IDE_IORDY[0:1] is released, and no wait is generated.

5) Device negates IDE_IORDY[0:1] before t_A. IDE_IORDY[0:1] is released prior to negation and may be asserted for no more than 5 ns before release. A wait is generated.

6) The cycle completes after IDE_IORDY[0:1] is reasserted. For cycles where a wait is generated and IDE_IOR[0:1] is asserted, the device places read data on IDE_DATA[15:0] for t_{RD} before asserting IDE_IORDY[0:1].

Figure 8-24. Register Transfer to/from Device Timing Diagram

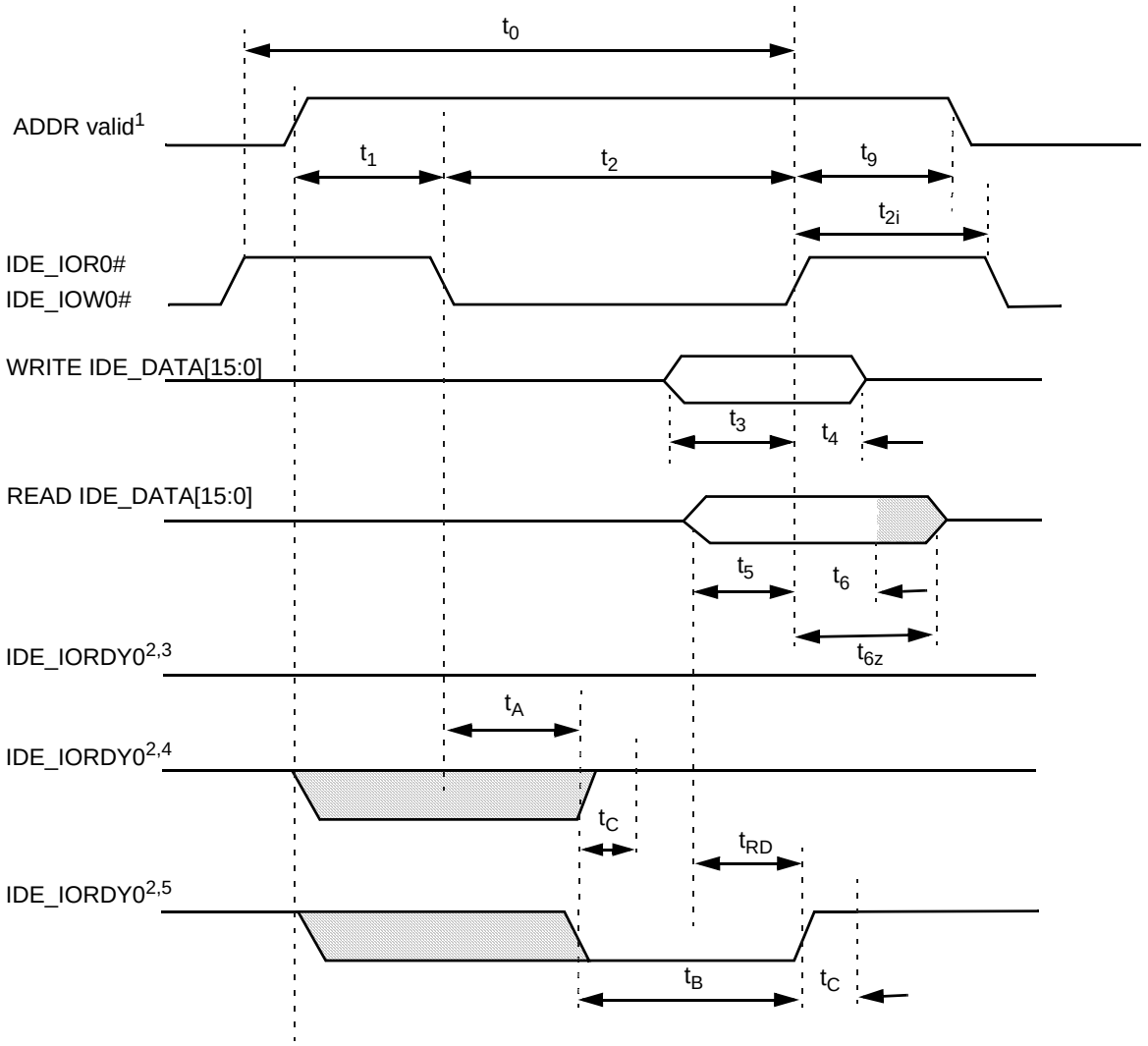
Electrical Specifications (Continued)

Table 8-25. PIO Data Transfer to/from Device Timing Parameters

Symbol	Parameter	Mode 0 (ns)	Mode 1 (ns)	Mode 2 (ns)	Mode 3 (ns)	Mode 4 (ns)
t_0	Cycle time[1] (min)	600	383	240	180	120
t_1	Address valid to IDE_IOR[0:1]#/IDE_IOW[0:1]# setup (min)	70	50	30	30	25
t_2	IDE_IOR[0:1]#/IDE_IOW[0:1]# 16-bit[1] (min)	165	125	100	80	70
t_{2i}	IDE_IOR[0:1]#/IDE_IOW[0:1]# recovery time[1] (min)	-	-	-	70	25
t_3	IDE_IOW[0:1]# data setup (min)	60	45	30	30	20
t_4	IDE_IOW[0:1]# data hold (min)	30	20	15	10	10
t_5	IDE_IOR[0:1]# data setup (min)	50	35	20	20	20
t_6	IDE_IOR[0:1]# data hold (min)	5	5	5	5	5
t_{6Z}	IDE_IOR[0:1]# data TRI-STATE[2] (max)	30	30	30	30	30
t_9	IDE_IOR[0:1]#/IDE_IOW[0:1]# to address valid hold (min)	20	15	10	10	10
t_{RD}	Read Data Valid to IDE_IORDY[0,1] active (if IDE_IORDY[0:1] initially low after t_A) (min)	0	0	0	0	0
t_A	IDE_IORDY[0:1] Setup time[3]	35	35	35	35	35
t_B	IDE_IORDY[0:1] Pulse Width (max)	1250	1250	1250	1250	1250
t_C	IDE_IORDY[0:1] assertion to release (max)	5	5	5	5	5

1. t_0 is the minimum total cycle time, t_2 is the minimum command active time, and t_{2i} is the minimum command recovery time or command inactive time. The actual cycle time equals the sum of the command active time and the command inactive time. The three timing requirements of t_0, t_2, and t_{2i} are met. The minimum total cycle time requirement is greater than the sum of t_2 and t_{2i}. (This means that a host implementation may lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data.)
2. This parameter specifies the time from the rising edge of IDE_IOR[0:1]# to the time that the data bus is no longer driven by the device (TRI-STATE).
3. The delay from the activation of IDE_IOR[0:1]# or IDE_IOW[0:1]# until the state of IDE_IORDY[0:1] is first sampled. If IDE_IORDY[0:1] is inactive, then the host waits until IDE_IORDY[0:1] is active before the PIO cycle is completed. If the device is not driving IDE_IORDY[0:1] negated after the activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_5 is met and t_{RD} is not applicable. If the device is driving IDE_IORDY[0:1] negated after the activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_{RD} is met and t_5 is not applicable.

Electrical Specifications (Continued)



Notes:

1) Device address consists of signals $\overline{\text{I}}$DE_CS[0:1]# and IDE_ADDR[2:0].

2) Negation of IDE_IORDY[0:1] is used to extend the PIO cycle. The determination of whether or not the cycle is to be extended is made by the host after t_A from the assertion of IDE_IOR[0:1]# or IDE_IOW[0:1]#.

3) Device never negates IDE_IORDY[0:1]. Devices keep IDE_IORDY[0:1] released, and no wait is generated.

4) Device negates IDE_IORDY[0:1] before t_A but causes IDE_IORDY[0:1] to be asserted before t_A. IDE_IORDY[0:1] is released, and no wait is generated.

5) Device negates IDE_IORDY[0:1] before t_A. IDE_IORDY[0:1] is released prior to negation and may be asserted for no more than 5 ns before release. A wait is generated.

6) The cycle completes after IDE_IORDY[0:1] is reasserted. For cycles where a wait is generated and IDE_IOR[0:1]# is asserted, the device places read data on IDE_DATA[15:0] for t_{RD} before asserting IDE_IORDY[0:1].

Figure 8-25. PIO Data Transfer to/from Device Timing Diagram

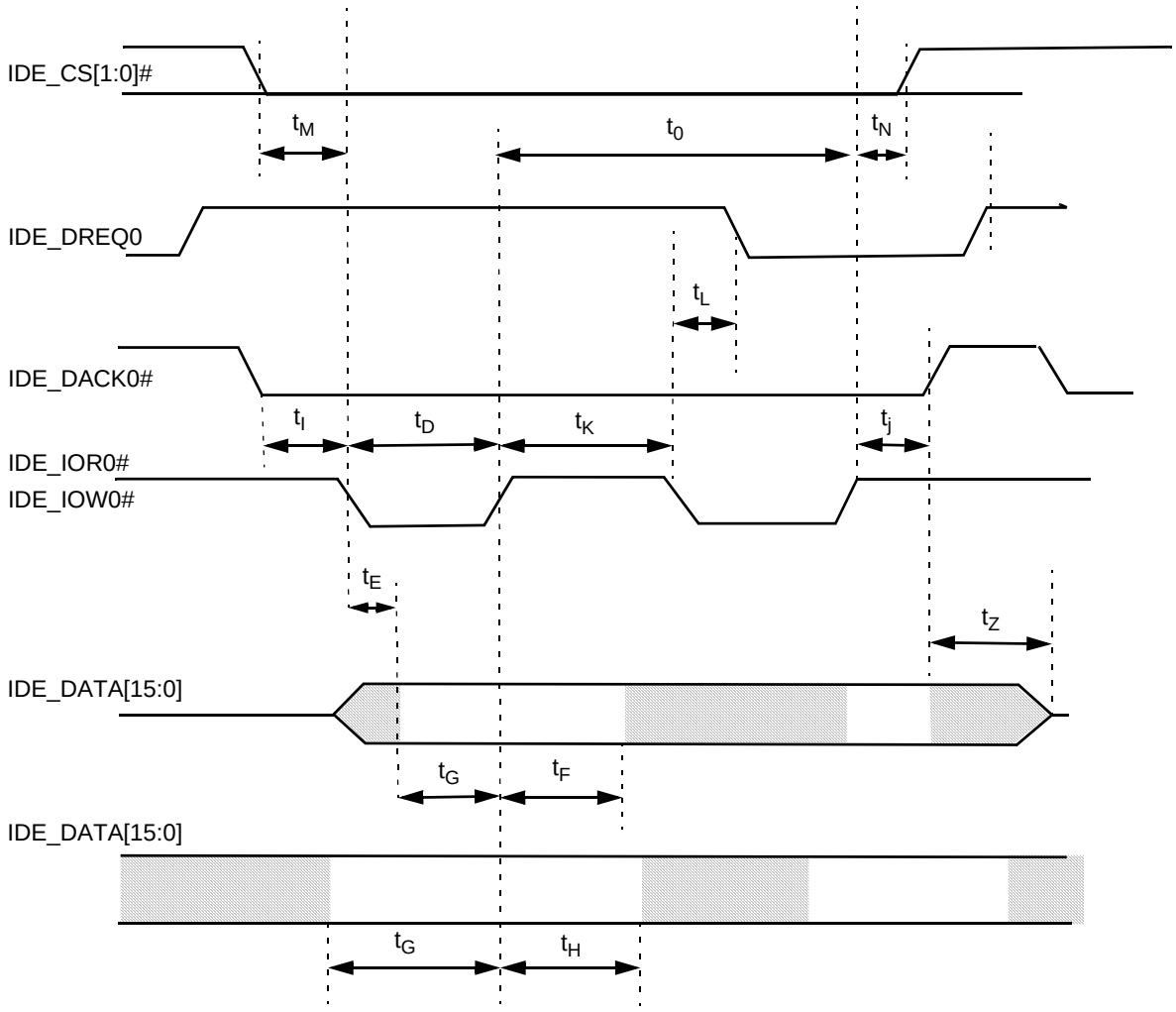
Electrical Specifications (Continued)

Table 8-26. Multiword DMA Data Transfer Timing Parameters

Symbol	Parameter	Mode 0 (ns)	Mode 1 (ns)	Mode 2 (ns)
t_0	Cycle time[1] (min)	480	150	120
t_D	IDE_IOR[0:1]#/IDE_IOW[0:1]# (min)	215	80	70
t_E	IDE_IOR[0:1]# data access (max)	150	60	50
t_F	IDE_IOR[0:1]# data hold (min)	5	5	5
t_G	IDE_IOW[0:1]#/IDE_IOW[0:1]# data setup (min)	100	30	20
t_H	IDE_IOW[0:1]# data hold (min)	20	15	10
t_I	IDE_DACK[0:1]# to IDE_IOR[0:1]#/IDE_IOW[0:1]# setup (min)	0	0	0
t_J	IDE_IOR[0:1]#/IDE_IOW[0:1]# to IDE_DACK[0:1]# hold (min)	20	5	5
t_{KR}	IDE_IOR[0:1]# negated pulse width (min)	50	50	25
t_{KW}	IDE_IOW[0:1]# negated pulse width (min)	215	50	25
t_{LR}	IDE_IOR[0:1]# to IDE_DREQ[0:1] delay (max)	120	40	35
t_{LW}	IDE_IOW[0:1]# to IDE_DREQ0,1 delay (max)	40	40	35
t_M	IDE_CS[0:1]# valid to IDE_IOR[0:1]#/IDE_IOW[0:1]# (min)	50	30	25
t_N	IDE_CS[0:1]# hold	15	10	10
t_Z	IDE_DACK[0:1]# to TRI-STATE	20	25	25

1. t_0 is the minimum total cycle time, t_D is the minimum command active time, and t_{KR} or t_{KW} is the minimum command recovery time or command inactive time. The actual cycle time equals the sum of the command active time and the command inactive time. The three timing requirements of t_0, t_D and $t_{KR/KW}$, are met. The minimum total cycle time requirement t_0 is greater than the sum of t_D and $t_{KR/KW}$. (This means that a host implementation can lengthen t_D and/or $t_{KR/KW}$ to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data.)

Electrical Specifications (Continued)



Notes:

1) For Multiword DMA transfers, the Device may negate IDE_DREQ[0:1] within the tL specified time once IDE_DACK[0:1 is asserted, and reassert it again at a later time to resume the DMA operation. Alternatively, if the device is able to co tinue the transfer of data, the device may leave IDE_DREQ[0:1] asserted and wait for the host to reasse IDE_DACK[0:1]#.

2) This signal can be negated by the host to Suspend the DMA transfer in process.

Figure 8-26. Multiword DMA Data Transfer Timing Diagram

Electrical Specifications (Continued)

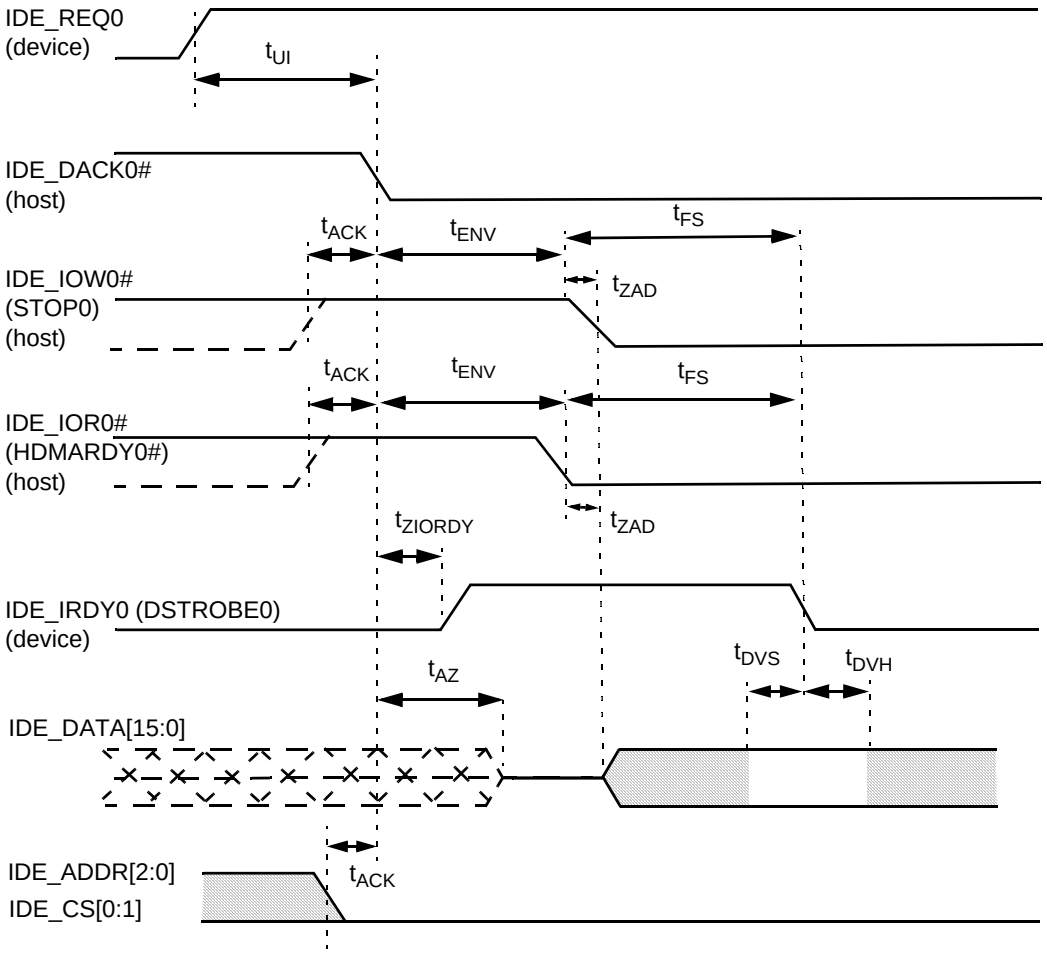
Table 8-27. UltraDMA Data Burst Timing Parameters

Symbol	Parameter	Mode 0 (ns)		Mode 1 (ns)		Mode 2 (ns)	
		Min	Max	Min	Max	Min	Max
t_{2CYC}	Typical sustained average two cycle time	240		160		120	
	Two cycle time allowing for clock variations (from rising edge to next rising edge or from falling edge to next falling edge of STROBE)	235		156		117	
t_{CYC}	Cycle time allowing for asymmetry and clock variations (from STROBE edge to STROBE edge)	114		75		55	
t_{DS}	Data setup time (at recipient)	15		10		7	
t_{DH}	Data hold time (at recipient)	5		5		5	
t_{DVS}	Data valid setup time at sender (from data bus being valid until STROBE edge)	70		48		34	
t_{DVH}	Data valid hold time at sender (from STROBE edge until data may become invalid)	6		6		6	
t_{FS}	First STROBE time (for device to first negate IDE_IRDY[0:1] (DSTROBE[0:1]) from IDE_IOW[0:1]# (STOP[0:1]) during a data in burst)	0	230	0	200	0	170
t_{LI}	Limited interlock time[1]	0	150	0	150	0	150
t_{MLI}	Interlock time with minimum[1]	20		20		20	
t_{UI}	Unlimited interlock time[1]	0		0		0	
t_{AZ}	Maximum time allowed for output drivers to release (from being asserted or negated)		10		10		10
t_{ZAH}	Minimum delay time required for output drivers to assert or negate (from released state)	20		20		20	
t_{ZAD}		0		0		0	
t_{ENV}	Envelope time (from IDE_DACK[0:1]# to IDE_IOW[0:1]# (STOP[0:1]) and IDE_IOR[0:1]# (HDMARDY[0:1]#) during data out burst initiation)	20	70	20	70	20	70
t_{SR}	STROBE to DMARDY time (if DMARDY# is negated before this long after STROBE edge, the recipient shall receive no more than one additional data WORD)		50		30		20
t_{RFS}	Ready-to-final-STROBE time (no STROBE edges shall be sent this long after negation of DMARDY#)		75		60		50
t_{RP}	Ready-to-pause time (time that recipient shall wait to initiate pause after negating DMARDY#)	160		125		100	
t_{IORDYZ}	Pull-up time before allowing IDE_IORDY[0:1] to be released		20		20		20
t_{ZIORDY}	Minimum time device shall wait before driving IDE_IORDY[0:1]	0		0		0	
T_{ACK}	Setup and hold times for IDE_DACK[0:1]# (before assertion or negation)	20		20		20	
T_{SS}	Time from STROBE edge to negation of IDE_DREQ[0:1] or assertion of IDE_IOW[0:1]# (STOP[0:1]) (when sender terminates a burst)	50		50		50	

1. t_{UI}, t_{MLI}, and t_{LI} indicate sender-to-recipient or recipient-to-sender interlocks, that is, one agent (either sender or recipient) is waiting for the other agent to respond with a signal before proceeding. t_{UI} is an unlimited interlock with no maximum time value. t_{MLI} is a limited time-out with a defined minimum. t_{LI} is a limited time-out with a defined maximum.

Electrical Specifications (Continued)

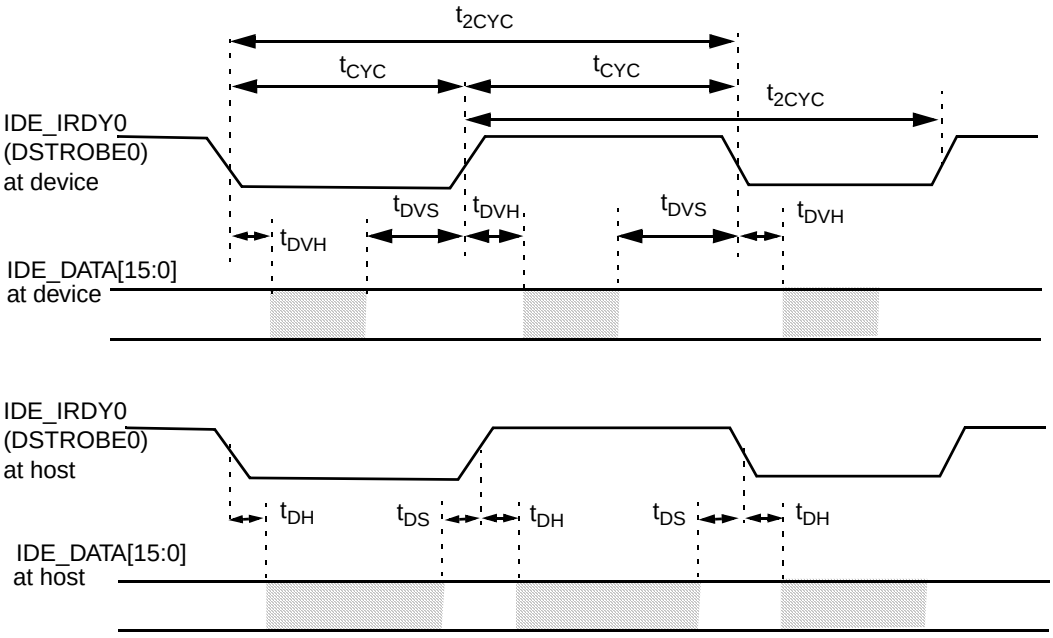
All timing parameters are measured at the connector of the device to which the parameter applies. For example, the sender stops generating STROBE edges t_{RFS} after the negation of DMARDY. Both STROBE and DMARDY timing measurements are taken at the connector of the sender.



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]), IDE_IOR[0:1]# (HDMARDY[0:1]#) and IDE_IRDY[0:1] (DSTROBE[0:1]) signal lines are not in effect until IDE_REQ[0:1] and IDE_DACK[0:1]# are asserted.

Figure 8-27. Initiating an UltraDMA Data in Burst Timing Diagram

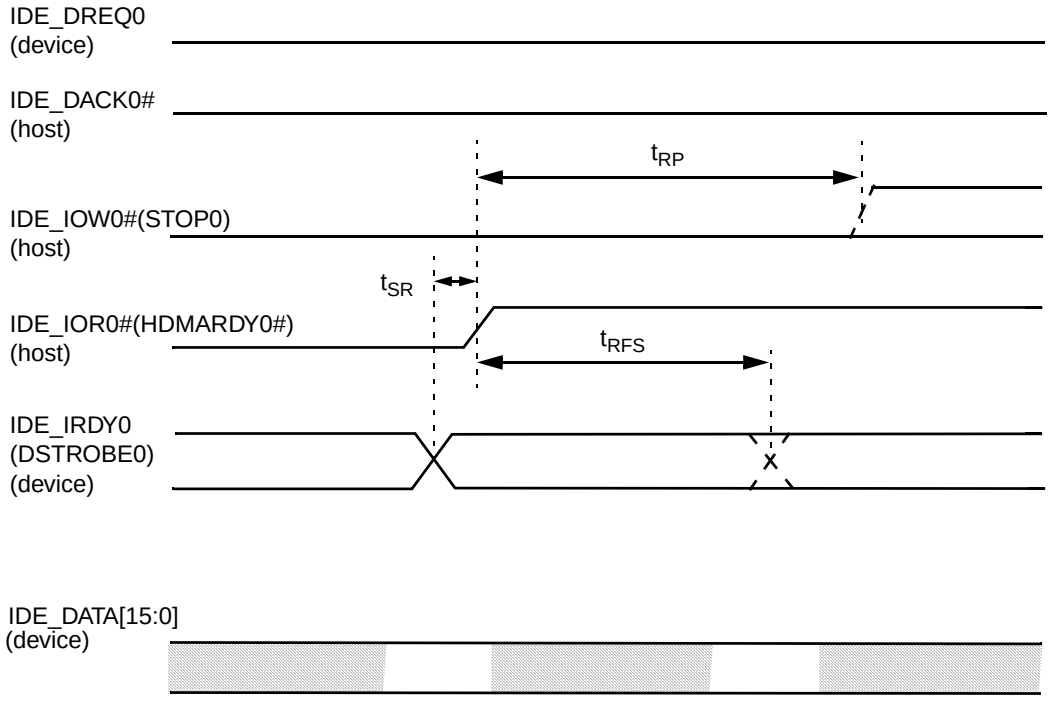
Electrical Specifications (Continued)



Note: IDE_DATA[15:0] and IDE_IRDY[0:1] (DSTROBE[0:1]) signals are shown at both the host and the device to emphasize that cable settling time and cable propagation delay do not allow the data signals to be considered stable at the host until a certain amount of time after they are driven by the device.

Figure 8-28. Sustained UltraDMA Data In Burst Timing Diagram

Electrical Specifications (Continued)



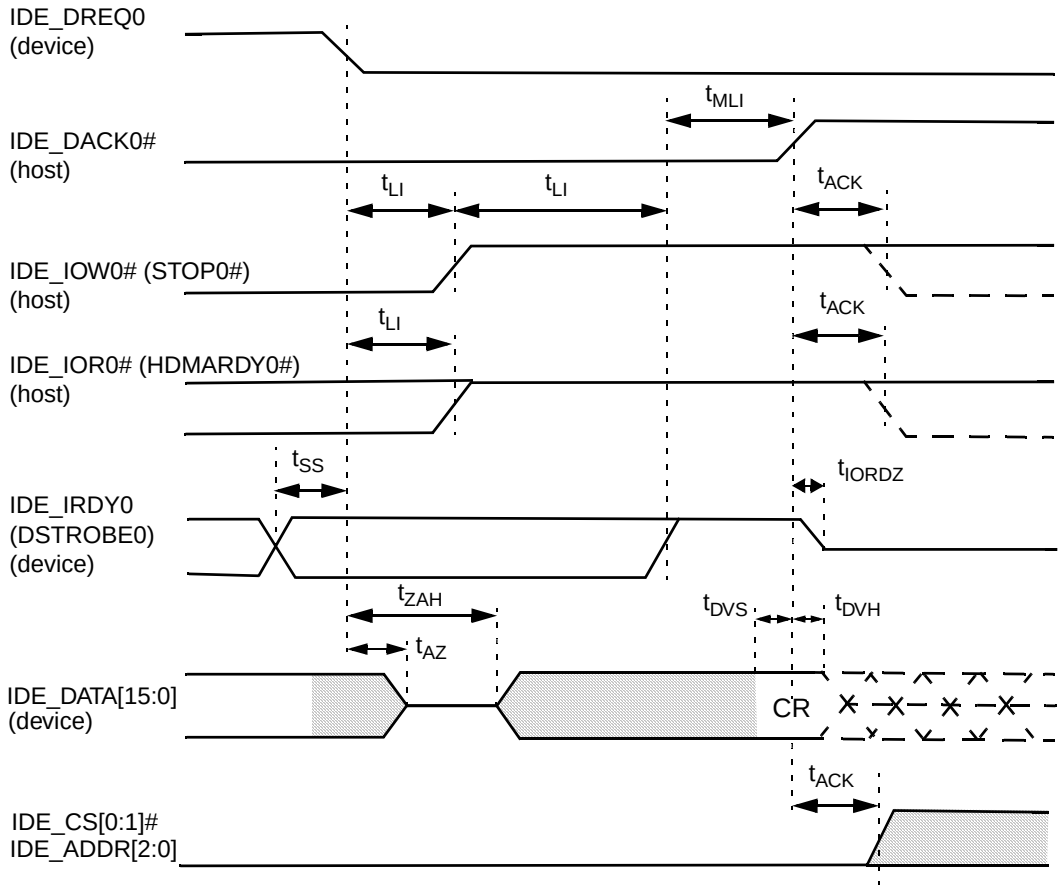
Notes:

1) The host can assert IDE_IOW[0:1]# (STOP[0:1]#) to request termination of the UltraDMA burst no sooner than t_{RP} after IDE_IOR[0:1]# (HDMARDY[0:1]#) is de-asserted.

2) If the t_{SR} timing is not satisfied, the host may receive up to two additional data WORDs from the device.

Figure 8-29. Host Pausing an UltraDMA Data In Burst Timing Diagram

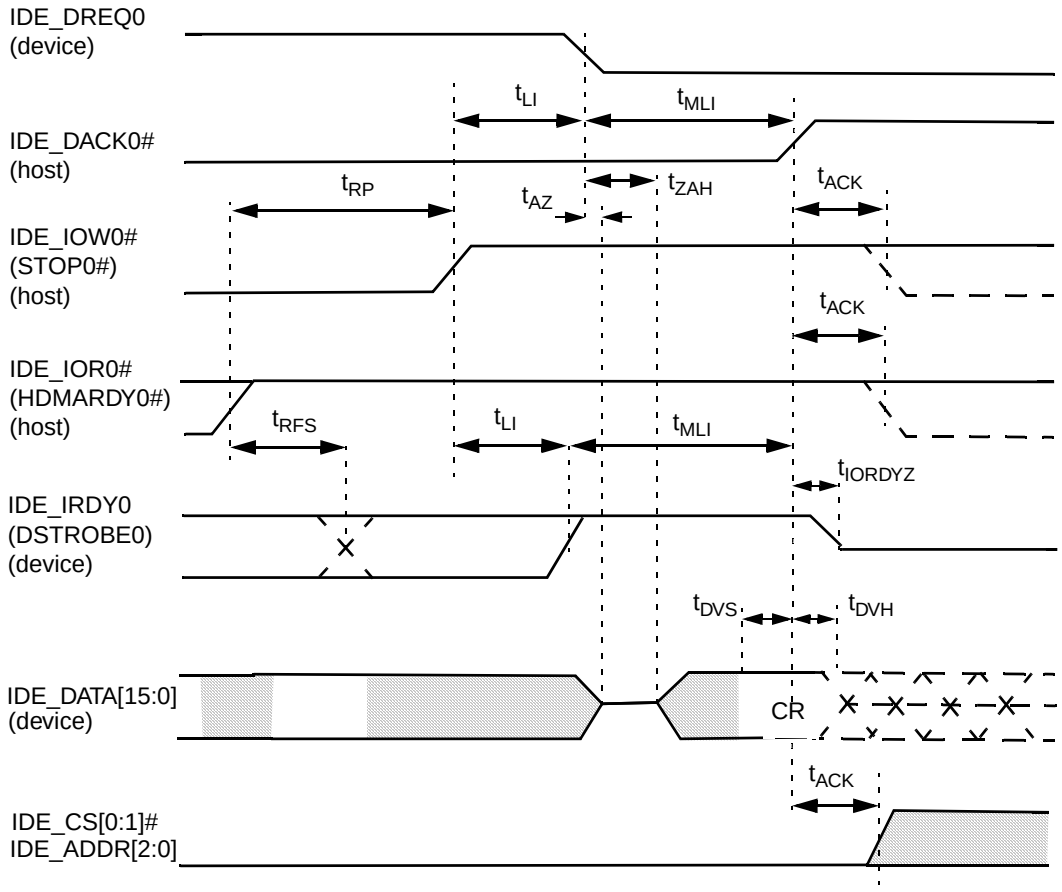
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IOR[0:1]# (HDMARDY[0:1]#), and IDE_IRDY[0:1] (DSTROBE[0:1]) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1]# are de-asserted.

Figure 8-30. Device Terminating an UltraDMA Data In Burst Timing Diagram

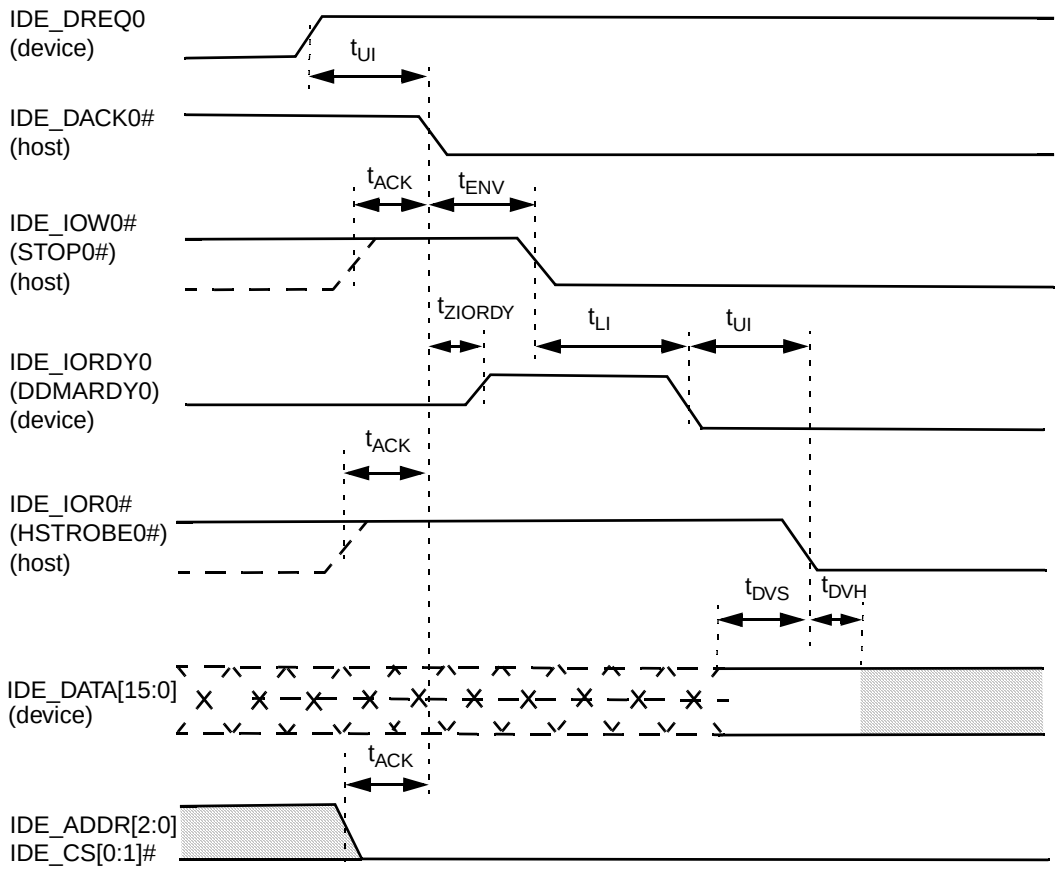
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IOR[0:1]# (HDMARDY[0:1]#), and IDE_IRDY[0:1] (DSTROBE[0:1]) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1] are de-asserted.

Figure 8-31. Host Terminating an UltraDMA Data In Burst Timing Diagram

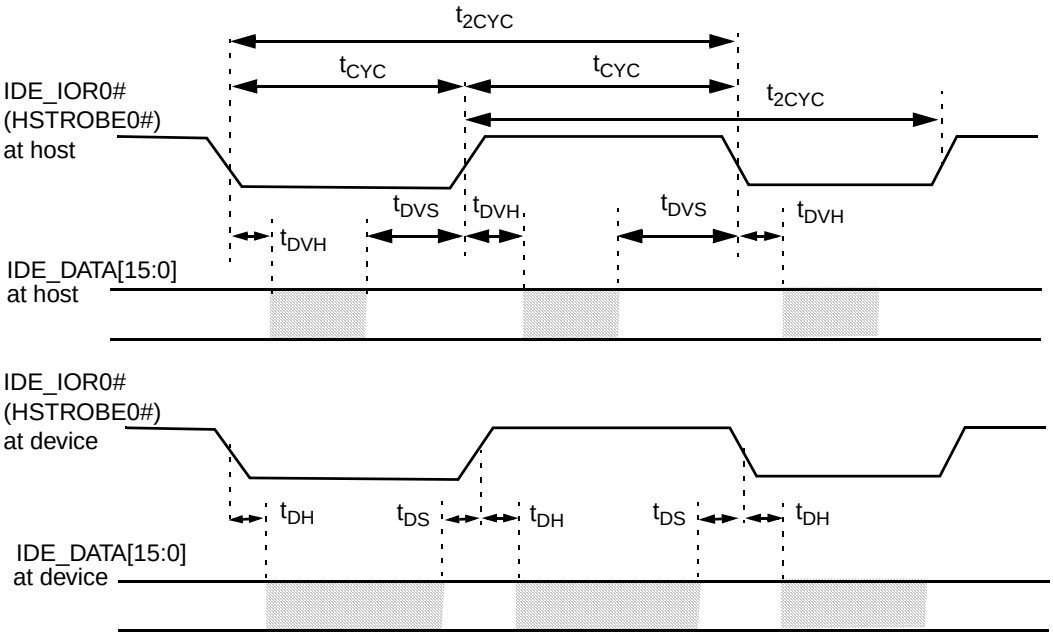
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]]# (STOP[0:1]#), IDE_IORDY[0:1]# (DDMARDY[0:1]) and IDE_IOR[0:1]# (HSTROBE[0:1]#) signal lines are not in effect until IDE_DREQ[0:1] and IDE_DACK[0:1]# are asserted.

Figure 8-32. Initiating an UltraDMA Data Out Burst Timing Diagram

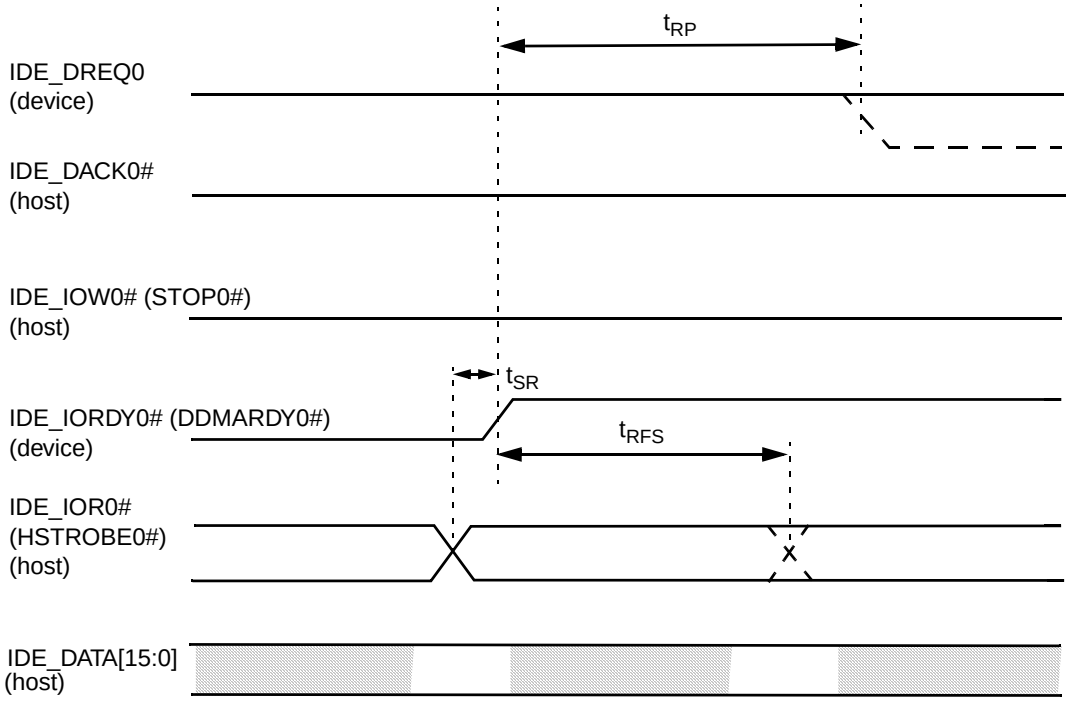
Electrical Specifications (Continued)



Note: IDE_DATA[15:0] and IDE_IOR[0:1]# (HSTROBE[0:1]#) signals are shown at both the device and the host to emphasize that cable settling time and cable propagation delay do not allow the data signals to be considered stable at the device until a certain amount of time after they are driven by the device.

Figure 8-33. Sustained UltraDMA Data Out Burst Timing Diagram

Electrical Specifications (Continued)



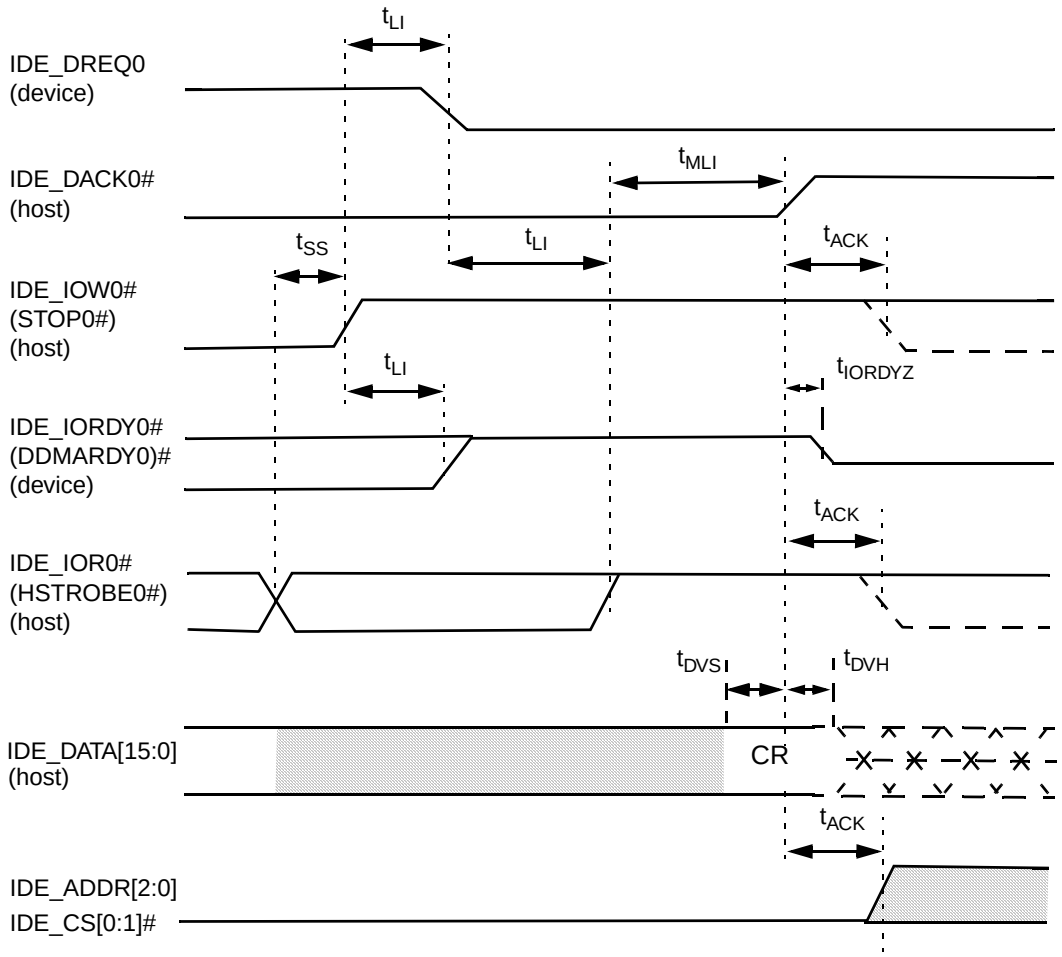
Notes:

1) The device can de-assert IDE_DREQ[0:1] to request termination of the UltraDMA burst no sooner than t_{RP} after IDE_IORDY[0:1]# (DDMARDY[0:1]#) is de-asserted.

2) If the t_{SR} timing is not satisfied, the device may receive up to two additional datawords from the host.

Figure 8-34. Device Pausing an UltraDMA Data Out Burst Timing Diagram

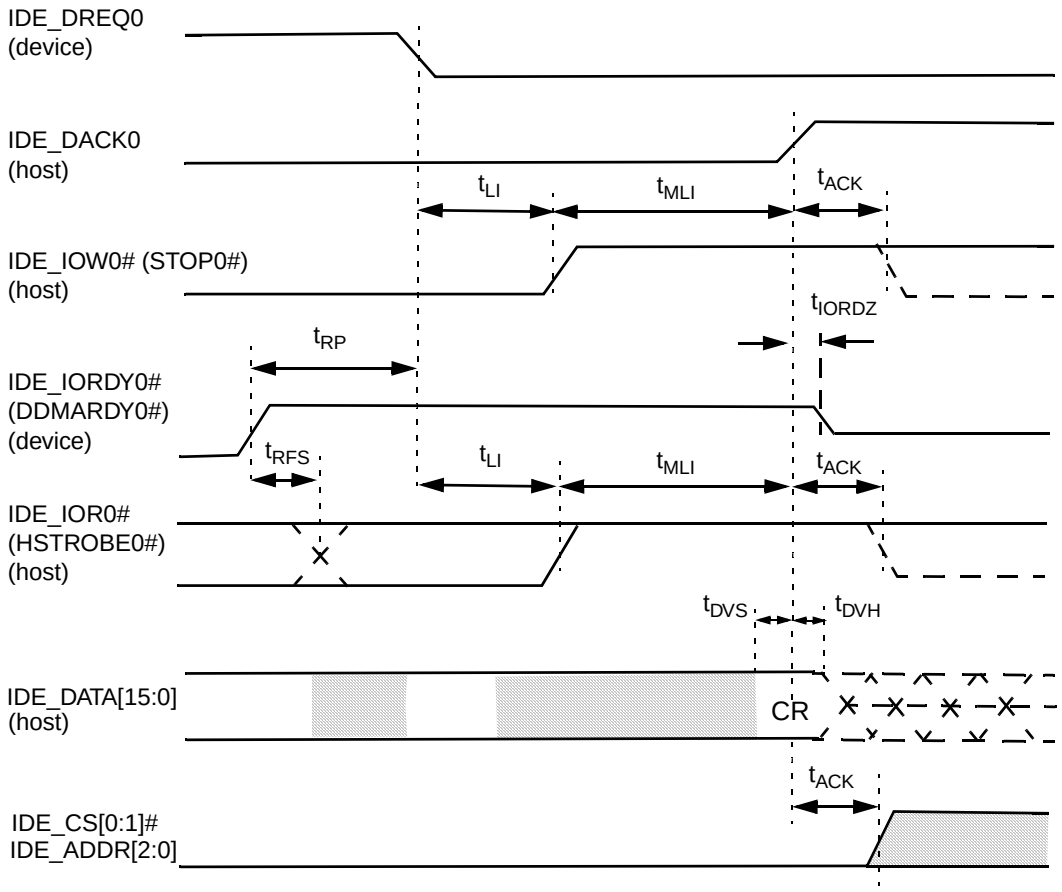
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IORDY[0,1]# (DDMARDY[0:1]#) and IDE_IOR[0:1]# (HSTROBE[0:1]#) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1]# are de-asserted.

Figure 8-35. Host Terminating an UltraDMA Data Out Burst Timing Diagram

Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IORDY[0:1]# (DDMARDY[0:1]#) and IDE_IOR[0:1]# (HSTROBE[0:1]#) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1]# are de-asserted.

Figure 8-36. Device Terminating an UltraDMA Data Out Burst Timing Diagram

Electrical Specifications (Continued)

8.3.9 Universal Serial Bus (USB) Interface

Table 8-28. USB Timing Parameters

Symbol	Parameter	Min	Max	Unit	Figure	Comments
Full Speed Source[1, 2]						
t_{USB_R1}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Rise Time	4	20	ns	8-37	(Monotonic) from 10% to 90% of the D_Port lines
t_{USB_F1}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Fall Time	4	20	ns	8-37	(Monotonic) from 90% to 10% of the D_Port lines
t_{USB_FRFM}	Rise/Fall time matching	90	110	%		
t_{USB_FSDR}	Full-speed data rate	11.97	12.03	Mbps		Average bit rate 12 Mbps ± 0.25%
t_{USB_FSF}	Full-speed frame interval	0.9995	1.0005	ms		1.0 ms ± 0.05%
t_{period_F}	Full-speed period between data bits	83.1	83.5	ns		Average bit rate 12 Mbps
$t_{USB\ DOR}$	Driver-output resistance	28	43	W		Steady-state drive
t_{USB_DJ11}	Source differential driver jitter[3, 4] for consecutive transition	–3.5	3.5	ns	8-38	
t_{USB_DJ12}	Source differential driver jitter[3, 4] for paired transitions	–4.0	4.0	ns	8-38	
t_{USB_SE1}	Source EOP width[4, 5]	160	175	ns	8-38	
t_{USB_DE1}	Differential to EOP transition skew[4, 5]	–2	5	ns	8-39	
t_{USB_RJ11}	Receiver data jitter tolerance[4] for consecutive transition	–18.5	18.5	ns	8-40	
t_{USB_RJ12}	Receiver data jitter tolerance[4] for paired transitions	–9	9	ns	8-40	
Full Speed Receiver EOP Width[4]						
t_{USB_RE11}	Must reject as EOP[5]		40	ns	8-39	
t_{USB_RE12}	Must accept as EOP[5]	82		ns	8-39	
Low Speed Source[1, 6]						
t_{USB_R2}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Rise Time	75	300[6]	ns	8-37	(Monotonic) from 10% to 90% of the D_Port lines
t_{USB_F2}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Fall Time	75	300[6]	ns	8-37	(Monotonic) from 90% to 10% of the D_Port lines
t_{USB_LRFM}	Low-speed Rise/Fall time matching	80	120	%		
t_{USB_LSDR}	Low-speed data rate	1.4775	1.5225	Mbps		Average bit rate 1.5 Mbps ± 1.5%
t_{PERIOD_L}	Low-speed period	0.657	0.677	µs		at 1.5 Mbps
t_{USB_DJD21}	Source differential driver jitter[4] for consecutive transactions	–75	75	ns		Host (downstream)
t_{USB_DJD22}	Source differential driver jitter[4] for paired transactions	–45	45	ns	8-38	Host (downstream)
t_{USB_DJU21}	Source differential driver jitter[4] for consecutive transaction	–95	95	ns	8-38	Function (downstream)

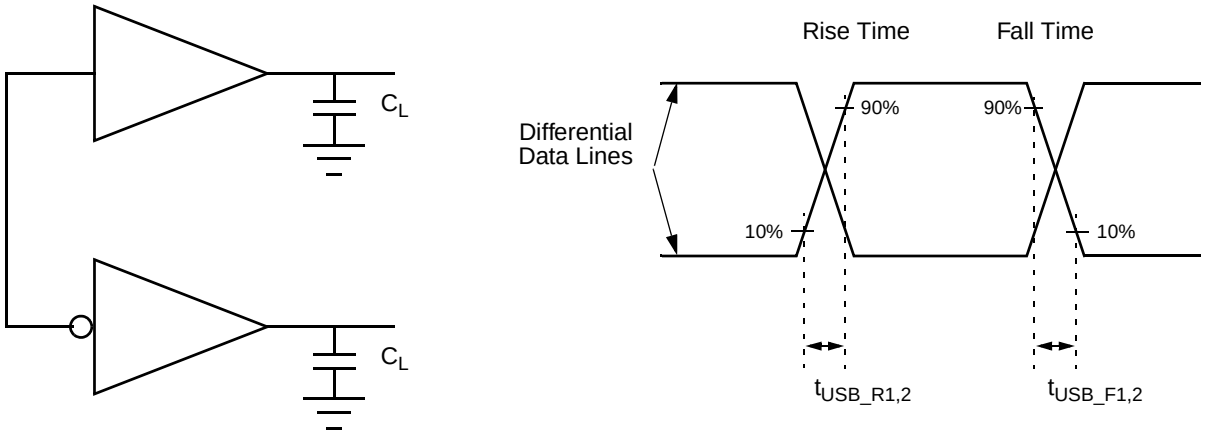
Electrical Specifications (Continued)

Table 8-28. USB Timing Parameters (Continued)

Symbol	Parameter	Min	Max	Unit	Figure	Comments
t_{USB_DJU22}	Source differential driver jitter[4] for paired transactions	–150	150	ns	8-38	Function (downstream)
t_{USB_SE2}	Source EOP width[4, 5]	1.25	1.5	µs	8-39	
t_{USB_DE2}	Differential to EOP [5] transition skew	–40	100	ns	8-39	
t_{USB_RJD21}	Receiver Data Jitter Tolerance[4] for consecutive transactions	–152	152	ns	8-40	Host (upstream)
t_{USB_RJD22}	Receiver Data Jitter Tolerance[4] for paired transactions	–200	200	ns	8-40	Host (upstream)
t_{USB_RJU21}	Receiver Data Jitter Tolerance[4] for consecutive transactions	–75	75	ns	8-40	Function (downstream)
t_{USB_RJU22}	Receiver Data Jitter Tolerance[4] for paired transactions	–45	45	ns	8-40	Function (downstream)
Low Speed Receiver EOP Width[5]						
t_{USB_RE21}	Must reject as EOP		330	ns	8-38	
t_{USB_RE22}	Must accept as EOP	675		ns	8-38	

1. Unless otherwise specified, all timings use a 50 pF capacitive load (C_L) to ground.
2. Full-speed timing has a 1.5 KΩ pull-up to 2.8 V on the DPOS_Port1,2,3 lines.
3. Timing difference between the differential data signals (DPOS_PORT1,2,3 and DNEG_PORT1,2,3).
4. Measured at the crossover point of differential data signals (DPOS_PORT1,2,3 and DNEG_PORT1,2,3).
5. EOP is the End of Packet where DPOS_PORT^t = DNEG_PORT = SE0. SE0 occurs when output level voltage ≤ V_{SE} (Min).
6. C_L = 350 pF.

Electrical Specifications (Continued)



Full Speed: 4 to 20 ns at C_L = 50 pF
Low Speed: 75 ns at C_L = 50 pF, 300 ns at C_L = 350 pF

Figure 8-37. Data Signal Rise and Fall Timing Diagram

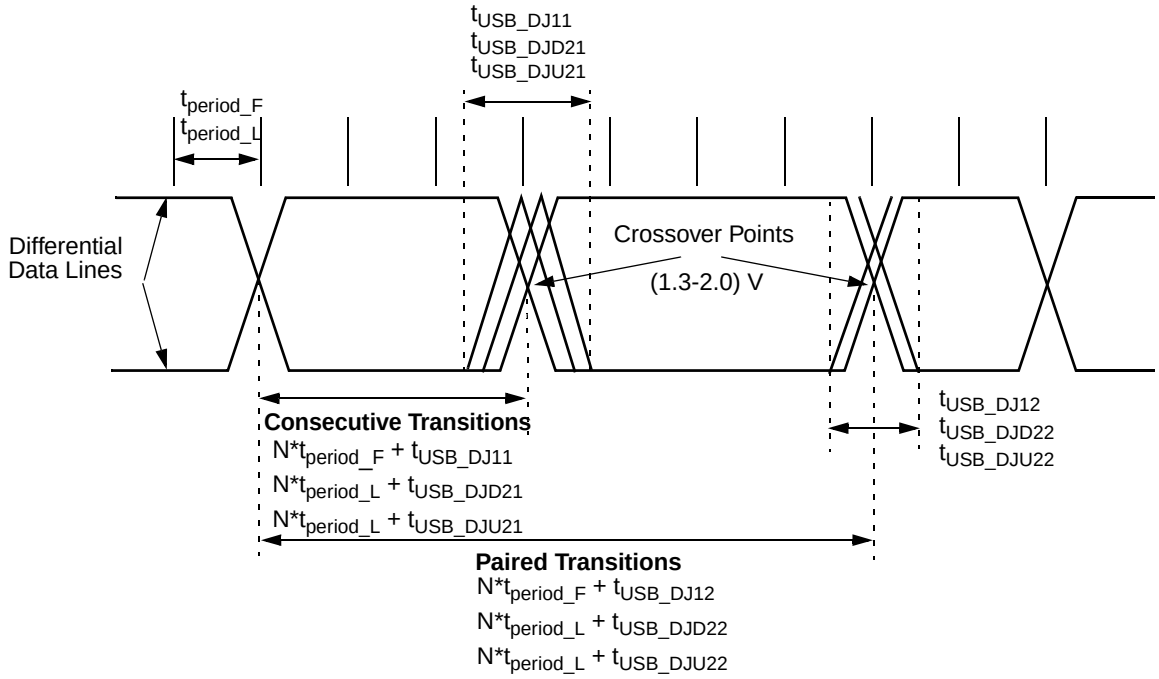


Figure 8-38. Source Differential Data Jitter Timing Diagram

Electrical Specifications (Continued)

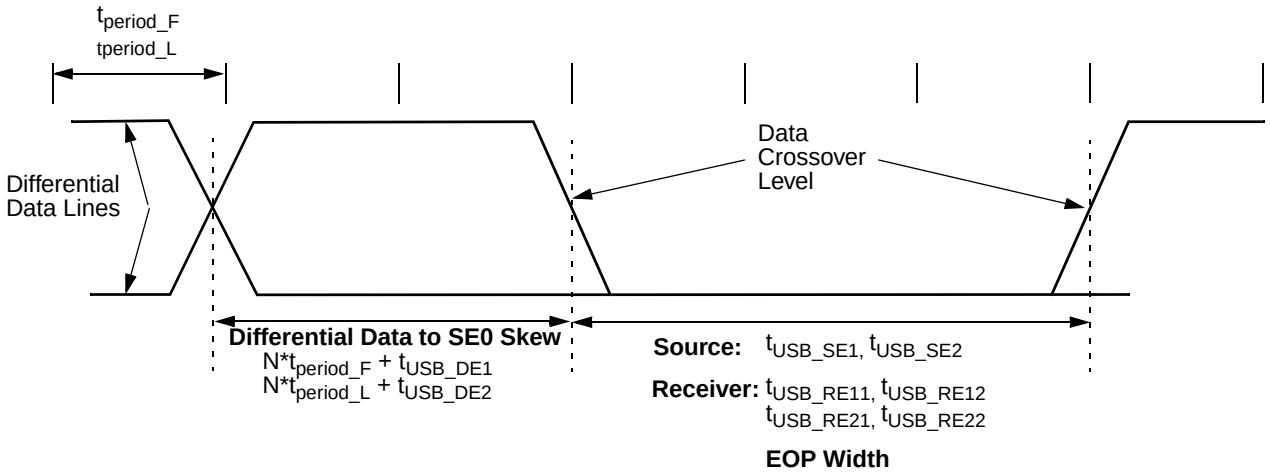


Figure 8-39. EOP Width Timing Diagram

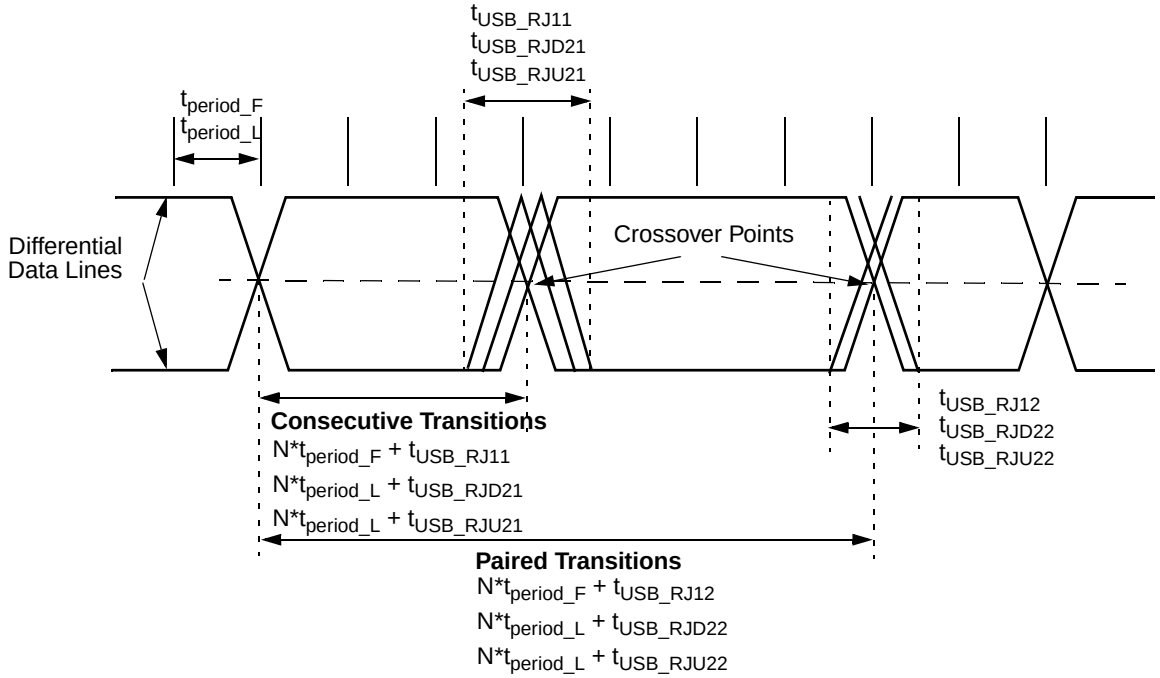


Figure 8-40. Receiver Jitter Tolerance Timing Diagram

Electrical Specifications (Continued)

8.3.10 Serial Port (UART)

Table 8-29. UART, Sharp-IR, SIR, and Consumer Remote Control Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{BT}	Single Bit Time in UART and Sharp-IR	t_{BTN} - 25 [1]	t_{BTN} + 25	ns	Transmitter
		t_{BTN} - 2%	t_{BTN} + 2%	ns	Receiver
t_{CMW}	Modulation Signal Pulse Width in Sharp-IR and Consumer Remote Control	t_{CWN} - 25 [2]	t_{CWN} + 25	ns	Transmitter
		500		ns	Receiver
t_{CMP}	Modulation Signal Period in Sharp-IR and Consumer Remote Control	t_{CPN} - 25 [3]	t_{CPN} + 25	ns	Transmitter
		t_{MMIN} [4]	t_{MMAX} [4]	ns	Receiver
t_{SPW}	SIR Signal Pulse Width	$(^3/_{16})$ x t_{BTN} - 15 [1]	$(^3/_{16})$ x t_{BTN} + 15 [1]	ns	Transmitter, Variable
		1.48	1.78	µs	Transmitter, Fixed
		1		µs	Receiver
S_{DRT}	SIR Data Rate Tolerance % of Nominal Data Rate		± 0.87%		Transmitter
			± 2.0%		Receiver
t_{SJT}	SIR Leading Edge Jitter % of Nominal Bit Duration		± 2.5%		Transmitter
			± 6.5%		Receiver

1. t_{BTN} is the nominal bit time in UART, Sharp-IR, SIR and Consumer Remote Control modes. It is determined by the setting of the Baud Generator Divisor registers.
2. t_{CWN} is the nominal pulse width of the modulation signal for Sharp-IR and Consumer Remote Control modes. It is determined by the MCPW field (bits [7:5]) of the IRTXMC register and the TXHSC bit (bit 2) of the RCCFG register.
3. t_{CPN} is the nominal period of the modulation signal for Sharp-IR and Consumer Remote Control modes. It is determined by the MCFR field (bits [4:0]) of the IRTXMC register and the TXHSC bit (bit 2) of the RCCFG register.
4. t_{MMIN} and t_{MMAX} define the time range within which the period of the incoming subcarrier signal has to fall in order for the signal to be accepted by the receiver. These time values are determined by the contents of register IRRXDC and the setting of the RXHSC bit (bit 5) of the RCCFG register .

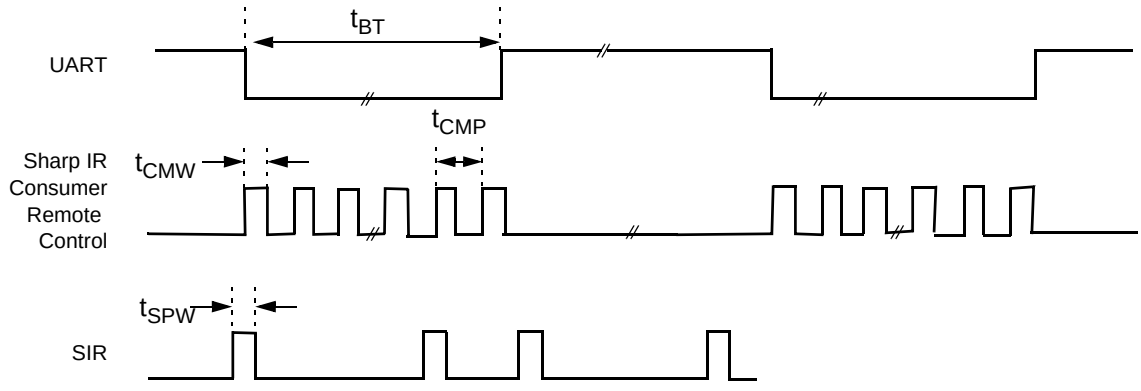


Figure 8-41. UART, Sharp-IR, SIR, and Consumer Remote Control Timing Diagram

Electrical Specifications (Continued)

8.3.11 Fast IR Port

Table 8-30. Fast IR Port Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{MPW}	MIR Signal Pulse Width	t_{MWN}-25[1]	t_{MWN}+25	ns	Transmitter
		60		ns	Receiver
M_{DRT}	MIR Transmitter Data Rate Tolerance		± 0.1%		
t_{MJT}	MIR Receiver Edge Jitter, % of Nominal Bit Duration		± 2.9%		
t_{FPW}	FIR Signal Pulse Width	120	130	ns	Transmitter
		90	160	ns	Receiver
t_{FDPW}	FIR Signal Double Pulse Width	245	255	ns	Transmitter
		215	285	ns	Receiver
F_{DRT}	FIR Transmitter Data Rate Tolerance		± 0.01%		
t_{FJT}	FIR Receiver Edge Jitter, % of Nominal Bit Duration		± 4.0%		

1. t_{MWN} is the nominal pulse width for MIR mode. It is determined by the M_PWID field (bits [4:0]) in the MIR_PW register at Offset 01h in bank 6 of logical device 5.

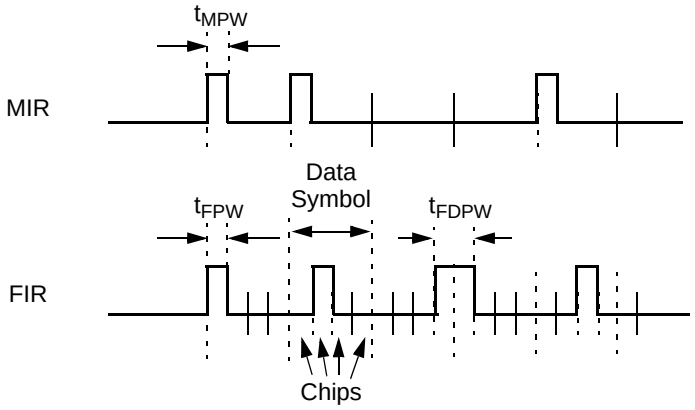


Figure 8-42. Fast IR (MIR and FIR) Timing Diagram

Electrical Specifications (Continued)

8.3.12 Parallel Port Interface

Table 8-31. Standard Parallel Port Timing Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{PDH}	Port Data Hold		500		ns	Times are system dependent and are therefore not tested.
t_{PDS}	Port Data Setup		500		ns	Times are system dependent and are therefore not tested.
t_{SW}	Strobe Width		500		ns	Times are system dependent and are therefore not tested.

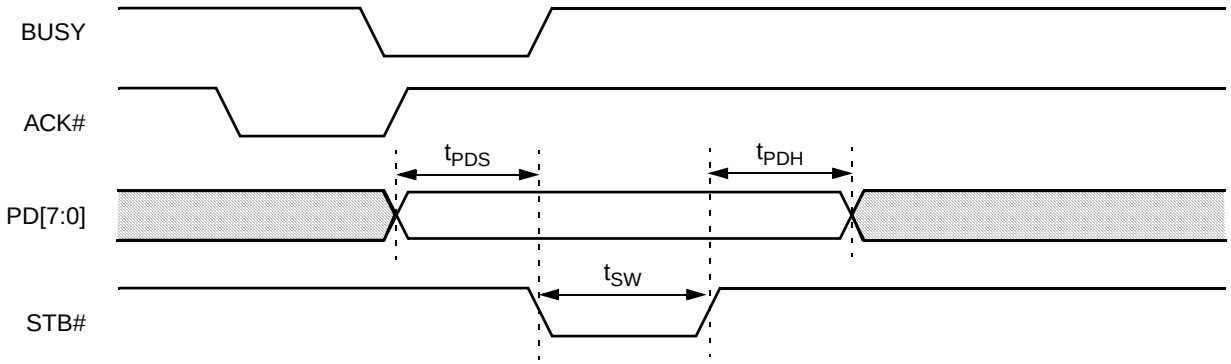


Figure 8-43. Standard Parallel Port Typical Data Exchange Timing Diagram

Electrical Specifications (Continued)

Table 8-32. Enhanced Parallel Port Timing Parameters

Symbol	Parameter	Min	Max	EPP 1.7	EPP 1.9	Unit
t_{WW19a}	WRITE# active from WAIT# low		45		x	ns
t_{WW19ia}	WRITE# inactive from WAIT# low		45		x	ns
t_{WST19a}	DSTRB# or ASTRB# active from WAIT# low		65		x	ns
t_{WEST}	DSTRB# or ASTRB# active after WRITE# active	10		x	x	ns
t_{WPDH}	PD[7:0] Hold after WRITE# inactive	0		x	x	ns
t_{WPDS}	PD[7:0] Valid after WRITE# active		15	x	x	ns
t_{EPDW}	PD[7:0] Valid width	80		x	x	ns
t_{EPDH}	PD[7:0] Hold after DSTRB# or ASTRB# inactive	0		x	x	ns

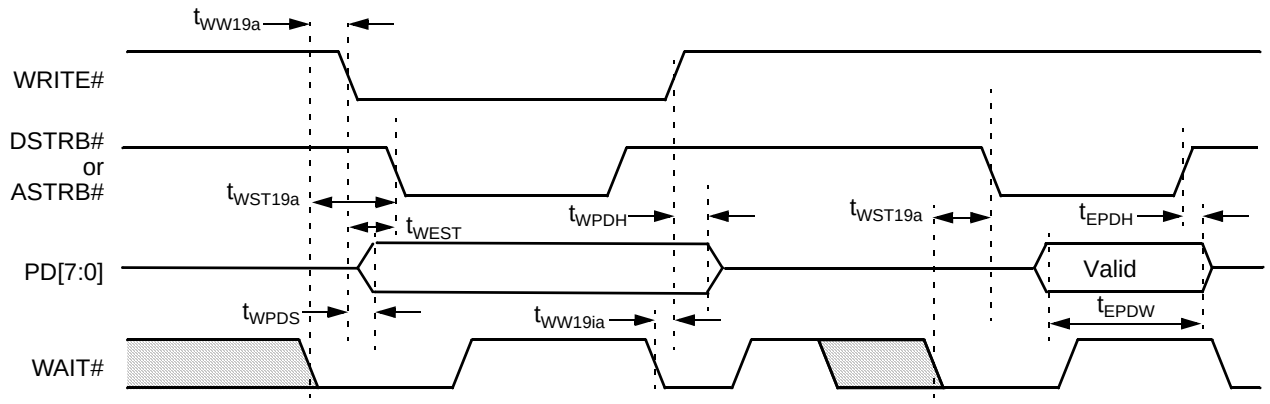


Figure 8-44. Enhanced Parallel Port Timing Diagram

Electrical Specifications (Continued)

8.3.12.1 Extended Capabilities Port (ECP)

Table 8-33. ECP Forward Mode Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{ECDSF}	Data Setup before STB# Active	0		ns	
t_{ECDHF}	Data Hold after BUSY Inactive	0		ns	
t_{ECLHF}	BUSY Active after STB# Active	75		ns	
t_{ECHHF}	STB# Inactive after BUSY Active	0	1	s	
t_{ECHLF}	BUSY Inactive after STB# Active	0	35	ms	
t_{ECLLF}	STB# Active after BUSY Inactive	0		ns	

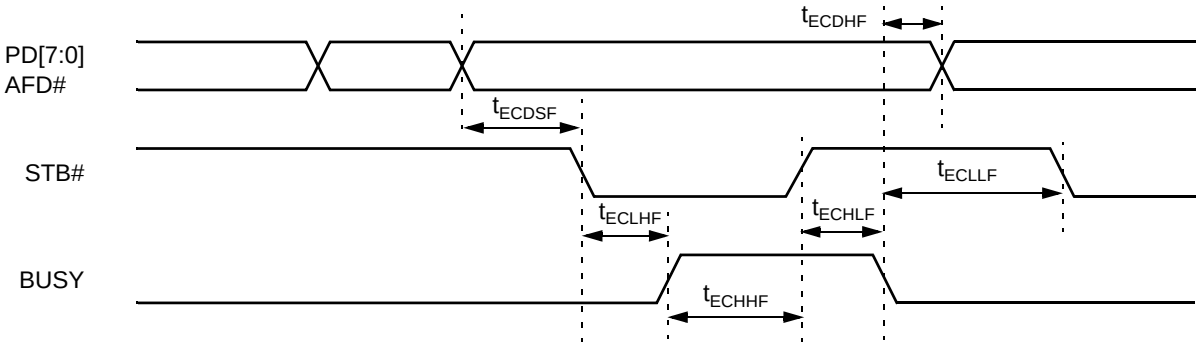


Figure 8-45. ECP Forward Mode Timing Diagram

Electrical Specifications (Continued)

Table 8-34. ECP Reverse Mode Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{ECDSR}	Data Setup before ACK# Active	0		ns	
t_{ECDHR}	Data Hold after AFD# Active	0		ns	
t_{ECLHR}	AFD# Inactive after ACK# Active	75		ns	
t_{ECHHR}	ACK# Inactive after AFD# Inactive	0	35	ms	
t_{ECHLR}	AFD# Active after ACK# Inactive	0	1	s	
t_{ECLLR}	ACK# Active after AFD# Active	0		ns	

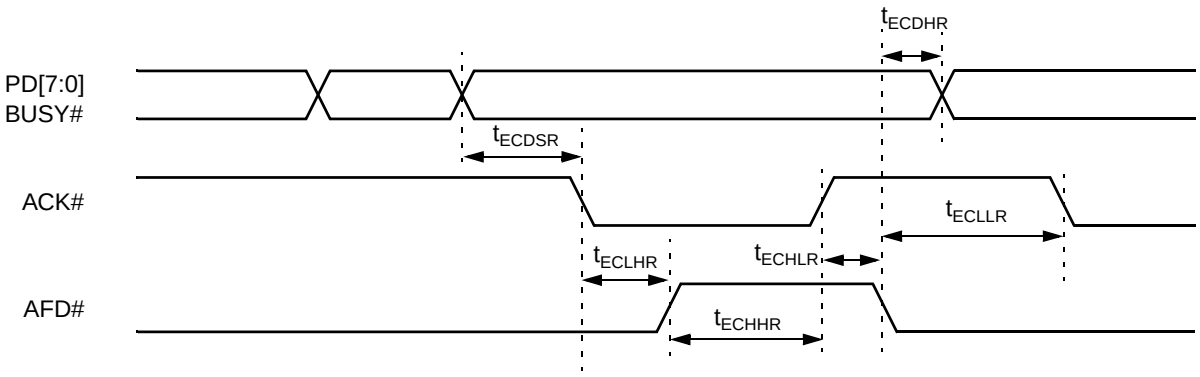


Figure 8-46. ECP Reverse Mode Timing Diagram

Electrical Specifications (Continued)

8.3.13 Audio Interface (AC97)

Table 8-35. AC97 Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{RST_LOW}	AC97_RST# active low pulse width	1.0			µs	
$t_{RST2CLK}$	AC97_RST# inactive to BIT_CLK startup delay	162.8			ns	

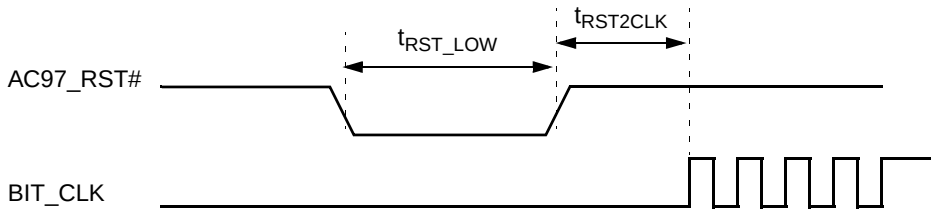


Figure 8-47. AC97 Reset Timing Diagram

Table 8-36. AC97 Sync Timing Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{SYNC_HIGH}	SYNC active high pulse width		1.3		µs	
t_{SYNC_IA}	SYNC inactive to BIT_CLK startup delay	162.8			ns	

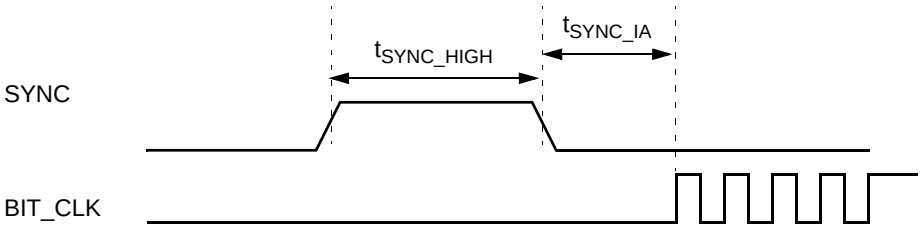


Figure 8-48. AC97 Sync Timing Diagram

Electrical Specifications (Continued)

Table 8-37. AC97 Clocks Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
F_{BIT_CLK}	BIT_CLK frequency		12.288		MHz	
t_{CLK_PD}	BIT_CLK period		81.4		ns	
t_{CLK_J}	BIT_CLK output jitter			750	ps	
t_{CLK_H}	BIT_CLK high pulse width[1]	32.56	40.7	48.84	ns	
t_{CLK_L}	BIT_CLK low pulse width[1]	32.56	40.7	48.84	ns	
F_{SYNC}	SYNC frequency		48.0		KHz	
t_{SYNC_PD}	SYNC period		20.8		µs	
t_{SYNC_H}	SYNC high pulse width		1.3		µs	
t_{SYNC_L}	SYNC low pulse width		19.5		µs	
F_{AC97_CLK}	AC97_CLK Frequency		24.576		MHz	
$t_{AC97_CLK_PD}$	AC97_CLK Period		40.7		ns	
$t_{AC97_CLK_D}$	AC97_CLK Duty Cycle	45		55	%	
$t_{AC97_CLK_FR}$	AC97_CLK Fall/Rise time	2		5	ns	
$t_{AC97_CLK_J}$	AC97_CLK output edge-to-edge jitter			100	ps	Measured from edge to edge

1. Worst case duty cycle restricted to 40/60.

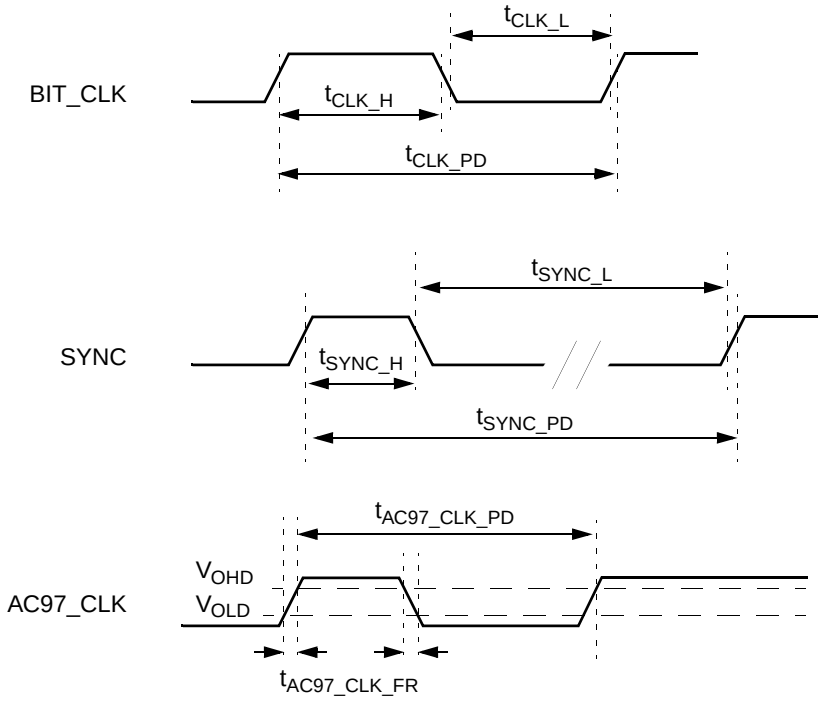


Figure 8-49. AC97 Clocks Diagram

Electrical Specifications (Continued)

Table 8-38. AC97 I/O Timing Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{AC97_S}	Input setup to falling edge of BIT_CLK	15.0			ns	
t_{AC97_H}	Hold from falling edge of BIT_CLK	10.0			ns	
t_{AC97_OV}	SDATA_OUT or SYNC valid after rising edge of BIT_CLK			15	ns	
t_{AC97_OH}	SDATA_OUT or SYNC hold time after falling edge of BIT_CLK	5			ns	
t_{AC97_SV}	Sync out valid after rising edge of BIT_CLK			15	ns	
t_{AC97_SH}	Sync out hold after falling edge of BIT_CLK	5			ns	

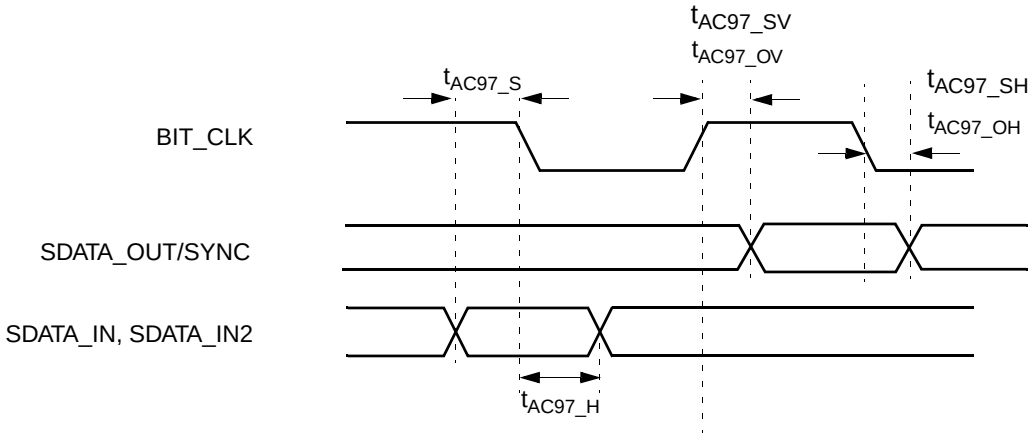


Figure 8-50. AC97 Data Timing Diagram

Electrical Specifications (Continued)

Table 8-39. AC97 Signal Rise and Fall Timing Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
$trise_{CLK}$	BIT_CLK rise time	2		6	ns	
$tfall_{CLK}$	BIT_CLK fall time	2		6	ns	
$trise_{SYNC}$	SYNC rise time	2		6	ns	C_L = 50 pF
$tfall_{SYNC}$	SYNC fall time	2		6	ns	C_L = 50 pF
$trise_{DIN}$	SDATA_IN rise time	2		6	ns	
$tfall_{DIN}$	SDATA_IN fall time	2		6	ns	
$trise_{DOUT}$	SDATA_OUT rise time	2		6	ns	C_L = 50 pF
$tfall_{DOUT}$	SDATA_OUT fall time	2		6	ns	C_L = 50 pF

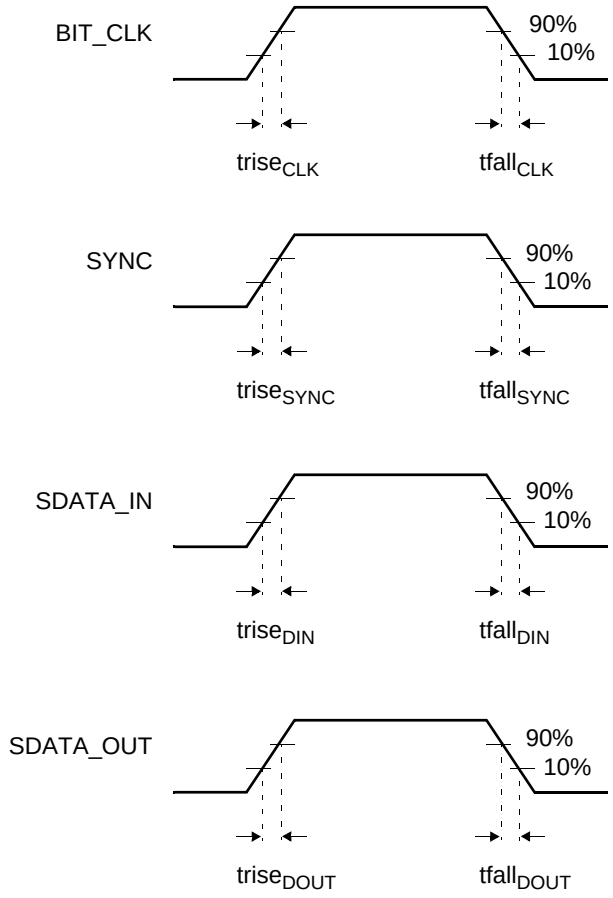
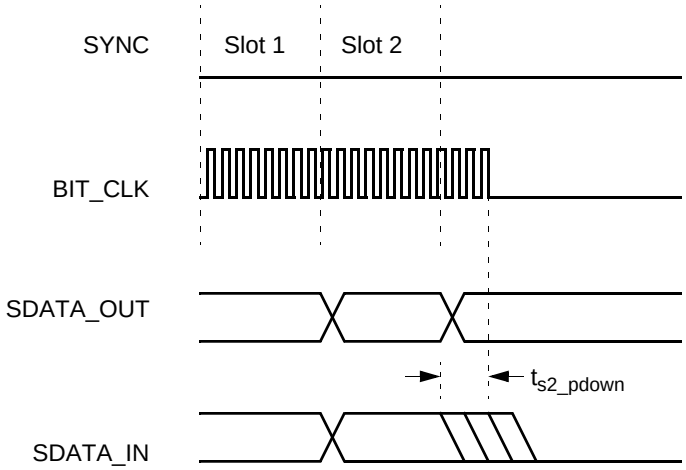


Figure 8-51. AC97 Rise and Fall Timing Diagram

Electrical Specifications (Continued)

Table 8-40. AC97 Low Power Mode Timing Parameters

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{s2_pdown}	End of Slot 2 to BIT_CLK, SDATA_IN low			1.0	µs	



Note: BIT_CLK is not to scale

Figure 8-52. AC97 Low Power Mode Timing Diagram

Electrical Specifications (Continued)

8.3.14 Power Management Interface

LED# Cycle time: 1 s ± 0.1 s, 40%-60% duty cycle.

Table 8-41. PWRBTN# Timing Parameters

Symbol	Parameter	Min	Max	Unit	Conditions
t_{PBTNP}	PWRBTN# Pulse Width[1]	16		ms	
t_{PBTNE}	Delay from PWRBTN# Events to ONCTL#	14	16	ms	

1. Not 100% tested.

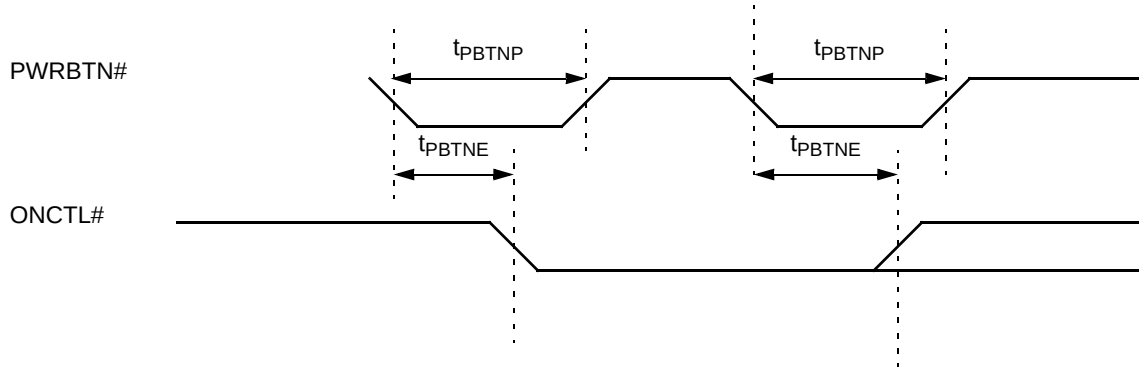


Figure 8-53. PWRBTN# Trigger and ONCTL# Timing Diagram

Table 8-42. Power Management Event (GPWIO) and ONCTL# Timing Parameters

Symbol	Parameter	Min	Max	Unit	Conditions
t_{PM}	Power Management Event to ONCTL# assertion		45	ns	

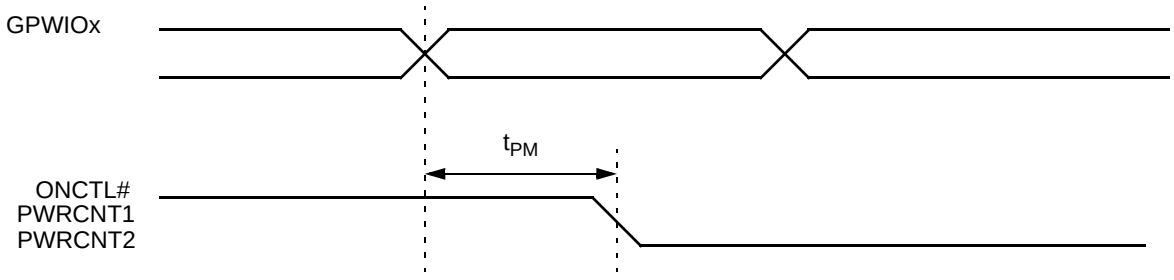


Figure 8-54. GPWIO and ONCTL# Timing Diagram

Electrical Specifications (Continued)

8.3.15 Power-Up Sequencing

Table 8-43. Power-Up Sequence Using the Power Button Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_1	Voltage sequence	-10	10	ms	
t_2	PWRBTN# inactive after V_{SB} or V_{SBL} applied, whichever is applied last	0	1	µs	PWRBTN# is an input and must be powered by V_{SB}.
t_3	PWRBTN# active pulse width	16	4000	ms	If PWRBTN# max is exceeded, ONCTL# will go inactive.
t_4	ONCTL# inactive after V_{SB} applied	0	1	ms	
t_5	Signal active after PWRBTN active	16	32	ms	
t_6	V_{CORE} and V_{IO} applied after ONCTL# active	0		ms	System determines when V_{CORE} and V_{IO} are applied, hence there is no maximum constraint.
t_7	POR# inactive after V_{CORE} and V_{IO} applied	50		ms	POR# must not glitch during active time.

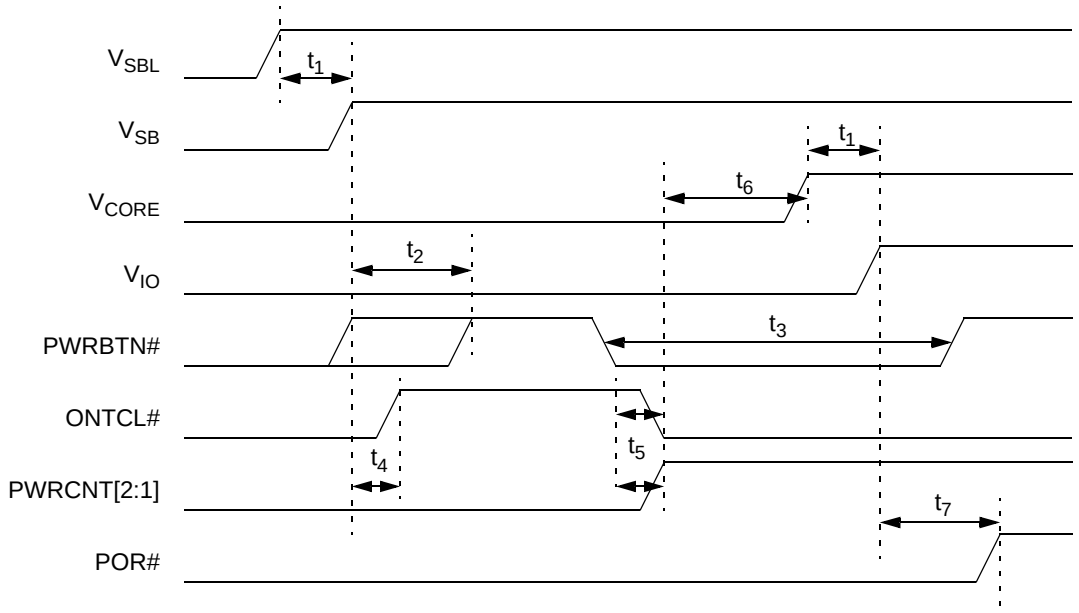
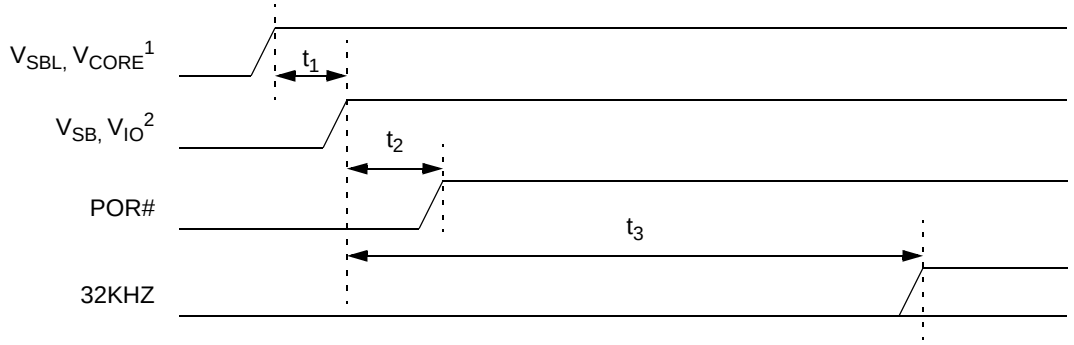


Figure 8-55. Power-Up Sequencing With PWRBTN# Timing Diagram

Electrical Specifications (Continued)

Table 8-44. Power-Up Sequence Not Using the Power Button Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_1	Voltage sequence	-10	10	ms	
t_2	POR# inactive after V_{SBL}, V_{CORE}, V_{SB}, and V_{IO} applied	50		ms	POR# must not glitch during active time.
t_3	32KHZ startup time		1	s	Time required for 32 KHz oscillator and 14.318 MHz derived from PLL6 to become stable at which time the RTC can reliably count.



1) V_{SBL} and V_{CORE} should be tied together.

2) V_{SB} and V_{IO} should be tied together.

Figure 8-56. Power-Up Sequencing Without PWRBTN# Timing Diagram

ACPI is non-functional when the power-up sequence does not include using the power button. If ACPI functionality is desired, the power button must be toggled. This can be done externally or internally. GPIO63 is internally connected to PWRBTN#. To toggle the power button with software, GPIO63 must be programmed as an output using the normal GPIO programming protocol (see Section 5.4.1.1 "GPIO Support Registers" on page 231). GPIO63 must be pulsed low for at least 16 ms and not more than 4 sec. Asserting POR# has no effect on ACPI. If POR# is asserted and ACPI was active prior to POR#, then ACPI will remain active after POR#. Therefore, BIOS must ensure that ACPI is inactive before GPIO63 is pulsed low.

Electrical Specifications (Continued)

8.3.16 JTAG Interface

Table 8-45. JTAG Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
	TCK Frequency (MHz)		25	MHz	
t_1	TCK Period	40		ns	
t_2	TCK High time	10		ns	
t_3	TCK Low time	10		ns	
t_4	TCK Rise time		4	ns	
t_5	TCK Fall Time		4	ns	
t_6	TDO Valid delay	3	25	ns	
t_7	Non-test outputs Valid delay	3	25	ns	50 pF load
t_8	TDO Float delay		30	ns	
t_9	Non-test outputs Float delay		36	ns	
t_{10}	TDI, TMS Setup time	8		ns	
t_{11}	Non-test inputs Setup time	8		ns	
t_{12}	TDI, TMS Hold time	7		ns	
t_{13}	Non-test inputs Hold time	7		ns	

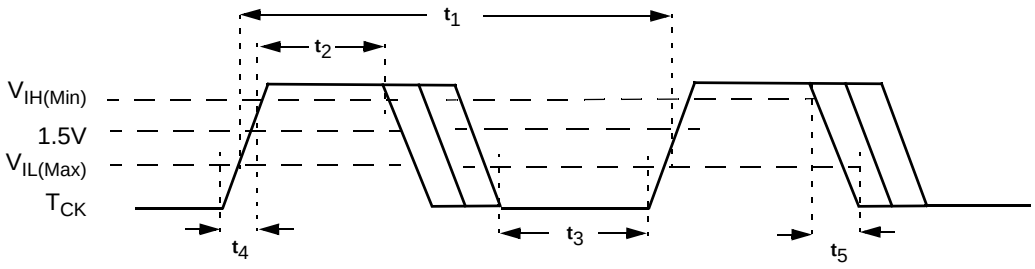


Figure 8-57. TCK Measurement Points Timing Diagram

Electrical Specifications (Continued)

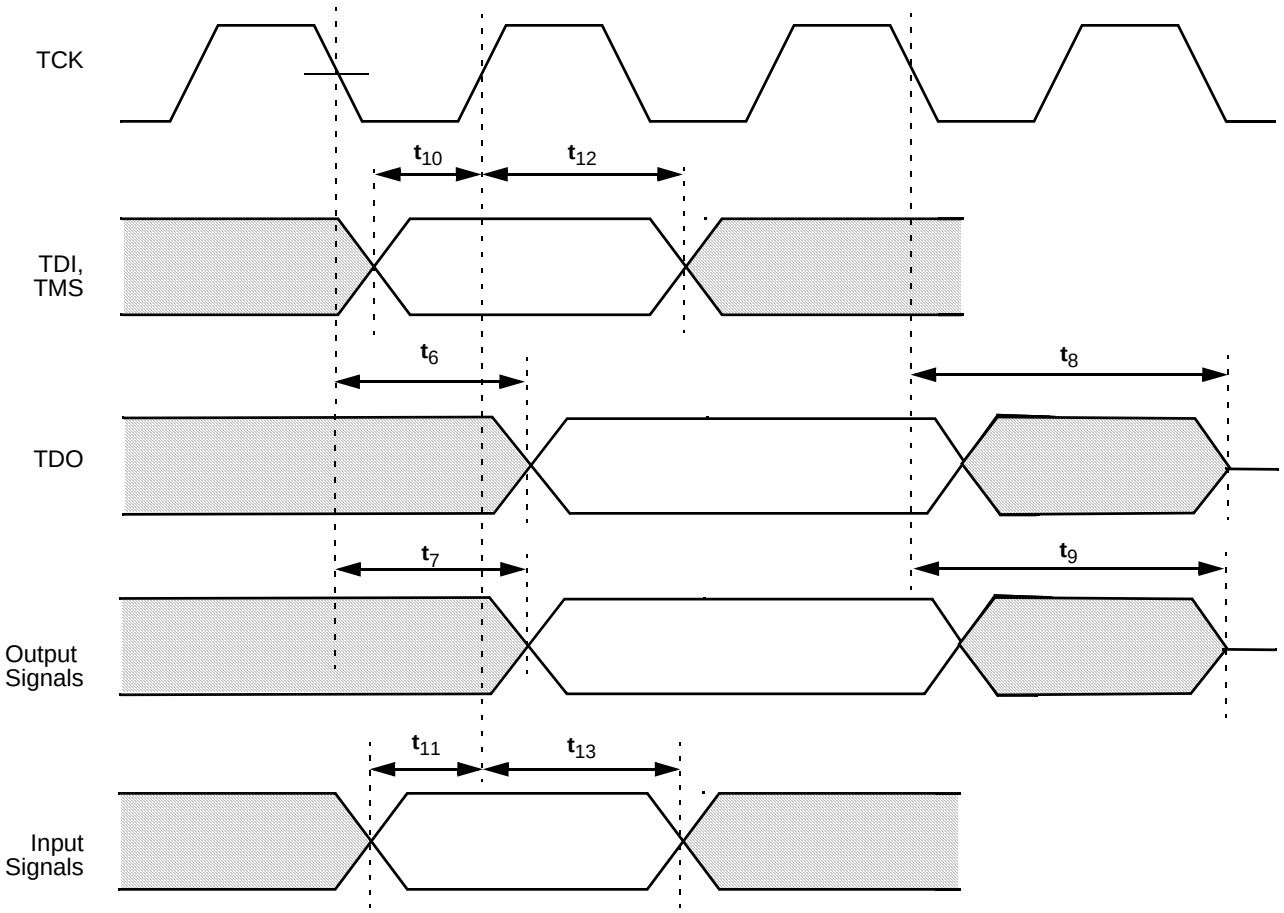
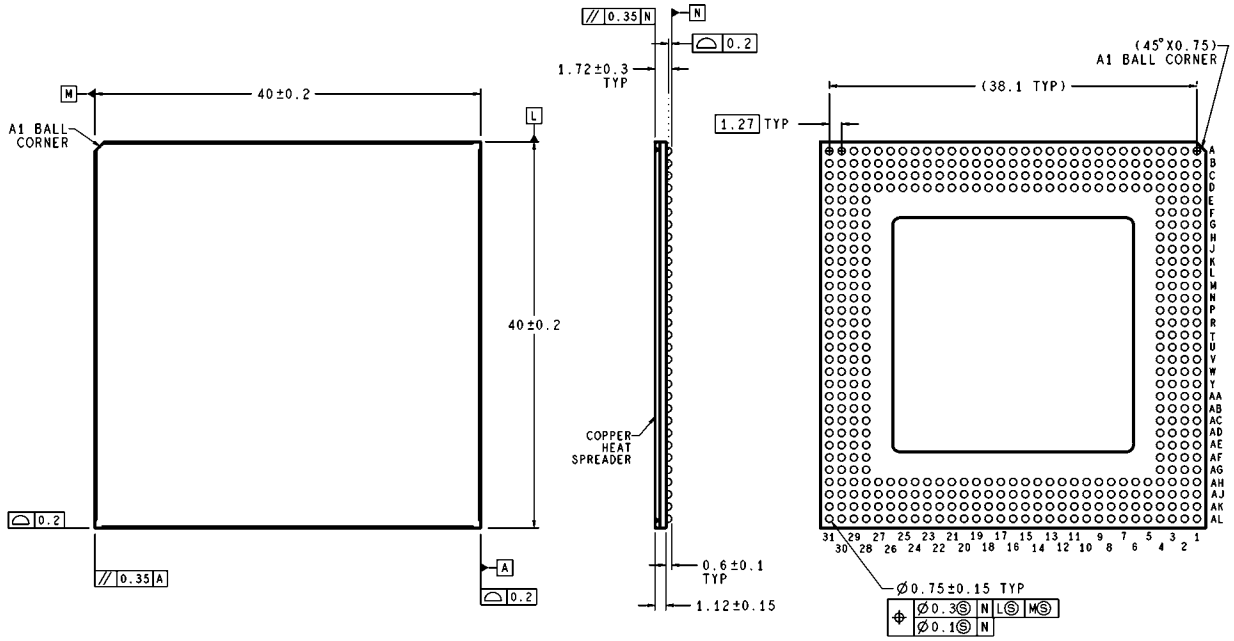


Figure 8-58. JTAG Test Timing Diagram

9.0 Package Specifications

The figures in this section provide the mechanical package outlines for the 432-Terminal EBGA (Enhanced Ball Grid Array) and 481-Terminal TEPBGA (Thermally Enhanced Ball Grid Array) packages.

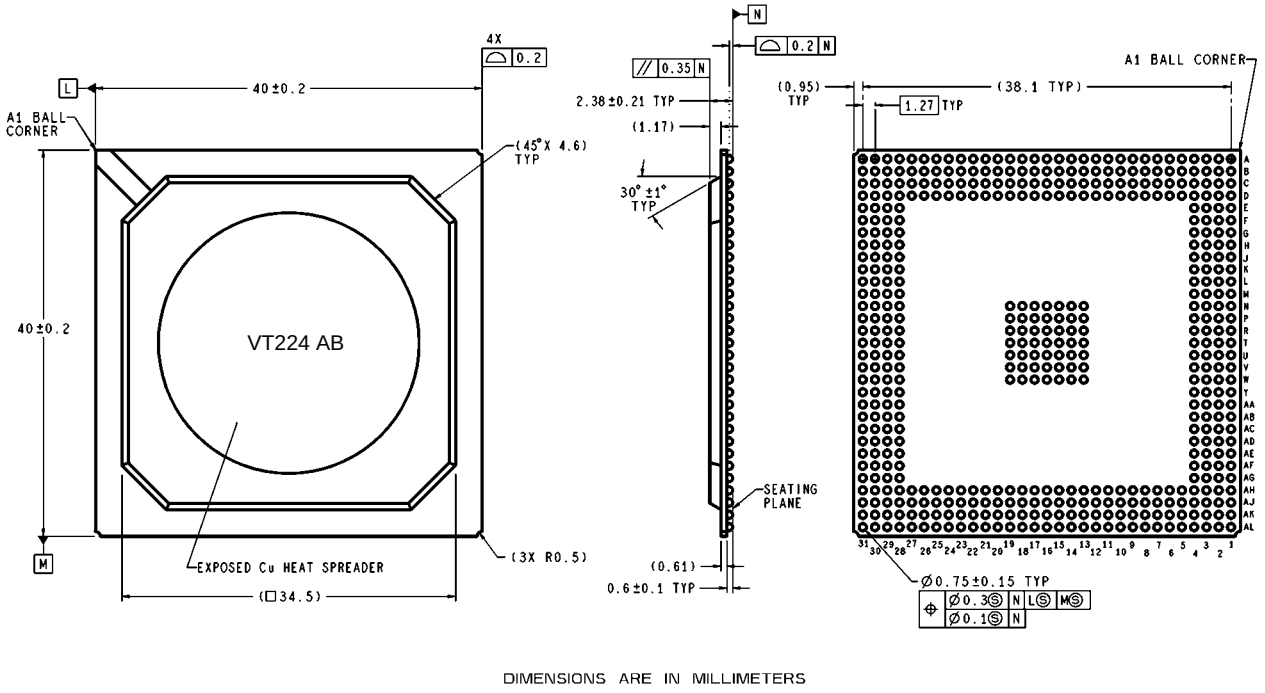


NOTES: UNLESS OTHERWISE SPECIFIED.

1) SOLDER BALL COMPOSITION: SN 63%, PB 37%.
2) DIMENSION IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM N.
3) REFERENCE JEDEC REGISTRATION MO-151, VARIATION -1.00, DATED JUNE 1997.
4) THETA JUNCTION TO CASE (T_{JC}) = 1°C/WATT.

Figure 9-1. 432-Terminal EBGA Package (Body Size: 40x40x1.72 mm; Pitch: 1.27 mm)

Package Specifications (Continued)



DIMENSIONS ARE IN MILLIMETERS

UFH481A (Rev A)

NOTES: UNLESS OTHERWISE SPECIFIED.

1) TEPBGA WITH LEAD (PB):
 a) SOLDER BALL COMPOSITION: SN 63%, PB 37%.
 b) SOLDERING PROFILE: 220^{O} C.
 c) PART IDENTIFICATION VIA LOT CODE. THE LAST TWO CHARACTERS OF THE SEVEN DIGIT CODE ARE: AA . . . AZ, BA . . . BZ, CA . . . CZ, . . . , QA . . . QZ.
2) TEPBGA LEAD (PB) FREE:
 a) SOLDER BALL COMPOSITION: SN 96.5%, AG 3.5%.
 b) SOLDERING PROFILE: 260^{O} C
 c) PART IDENTIFICATION VIA LOT CODE. THE LAST TWO CHARACTERS OF THE SEVEN DIGIT CODE ARE: RA . . . RZ, SA . . . SZ, TA . . . TZ, . . . , ZA . . . ZZ.
3) DIMENSION IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM N.
4) THE MOLD SURFACE AREA MAY INCLUDE DIMPLE FOR A1 BALL CORNER IDENTIFICATION.
5) REFERENCE JEDEC REGISTRATION MS-034, VARIATION BAU-1.
6) THETA JUNCTION TO CASE (T_{JC}) = 5°C/WATT.

Figure 9-2. 481-Terminal TEPBGA Package (Body Size: 40x40x2.38 mm; Pitch: 1.27 mm)

Appendix A Support Documentation

A.1 ORDER INFORMATION

Order Number (NSID)	Part Marking	Core Frequency (MHz)	Core Voltage (V_{CORE})	Temperature (Degree C)	Package
SC2200UCL-266	SC2200UCL-266	266	1.8V	0 - 85	EBGA
SC2200UCL-300	SC2200UCL-300	300	2.1V	0 - 85	
SC2200UFH-233	SC2200UFH-233	233	1.8V	0 - 85	TEPBGA[1]
SC2200UFH-266	SC2200UFH-266	266			
SC2200UFH-233B[2]	SC2200UFH-233B	233			
SC2200UFH-266B[2]	SC2200UFH-266B	266			
SC2200UFH-300	SC2200UFH-300	300	2.1		

1. A leadfree package version of this device is available. Use Spec. Flow NOPB to order the leadfree package. See Section 9.0 "Package Specifications" for how to identify the part.
2. Refer to I_{BAT} in Table 8-6 on page 362 for details.

A.2 DATASHEET REVISION HISTORY

This document is a report of the revision/creation process of the datasheet for the Geode SC2200. Any revisions (i.e., additions, deletions, parameter corrections, etc.) are recorded in the table below.

Table A-1. Revision History

Revision # (PDF Date)	Revisions / Comments
0.1 (August 1999)	First draft of datasheet. (For internal review only)
0.2 (October 1999)	Second draft.
0.5 (January 2000)	Draft.
0.8 (March 2000)	Draft
1.0 (July 2000)	Preliminary datasheet.
1.1 (August 2000)	Corrected typos, removed unimplemented features.
1.32 (February 2001)	Preliminary datasheet. GNT[1:0]# strapping functions changed. Minor modifications and corrections. Video Input Port (VIP) added.
2.0 (April 2002)	The entire datasheet has been reformatted and several sections re-written. Strap information and TEPBGA ball assignments have been added. Additionally, internal test signals and multiplexing have been included. Datasheet should be viewed as a new document.
2.1 (June 2002)	Release for posting on external web site. Changes made to the Architecture Overview, Signal Definitions, Core Logic Module, Video Processor Module, Electrical Specifications, and Package Specifications chapters.
3.0 (August 2002)	Major corrections include fixing TEPBGA ball numbers in “Two-Signal/Group Multiplexing” table (Table 2-7) and GPIO signal descriptions (Section 2.4.16). Additions include breaking out 233/266 and 300 MHz values for V_{CORE}, V_{SBL}, and V_{CCCRT} in “Operating Conditions” table (Table 8-2).
3.1 (February 2003)	Many minor changes mostly to the Video Processor and Electrical sections. Expounded on the notes in the Mechanical section.

Support Documentation (Continued)

Table A-1. Revision History (Continued)

Revision # (PDF Date)	Revisions / Comments
4.0 (March 2003)	Many edits. Changed all references to XpressAUDIO to audio.
4.1 (May 2003)	Changed V_{CORE} voltage for 300 MHz from 2.0V to 2.1V throughout manual. Other changes to the Electrical section. See Table A-2 "Edits to Current Revision" for details.

Table A-2. Edits to Current Revision

Section	Revision
General Description	• No changes.
Features	• Corrected TFT Interface feature for 800x600 non-interlaced TFT @ 16 bpp graphics: changed from up tp 75 Hz to up to 85 Hz.
Section 1.0 "Architecture Overview"	• No changes.
Section 2.0 "Signal Definitions"	• No changes.
Section 3.0 "General Configuration Block"	• No changes.
Section 4.0 "SuperI/O Module"	• No changes.
Section 5.0 "Core Logic Module"	• No changes.
Section 6.0 "Video Processor Module"	• No changes.
Section 7.0 "Debugging and Monitoring"	• No changes.
Section 8.0 "Electrical Specifications"	• Changed Min, Typ, and Max values for 300 MHz in Table 8-2 on page 359. • Changed V_{CORE} voltage for 300 MHz from 2.0V to 2.1V throughout manual. • Changed Typ Avg and Abs Max values for I_{COREON}, 300 MHz in Table 8-5 on page 361. • Table 8-6 on page 362: — Changed Typ value for $I_{COREIDLE}$, 300 MHz from 450 to 470. — Added SC2200UFH-300 to Footnote 4. • Deleted unnecessary text across of SDATA_OUT in Figure 8-52 on page 423.
Section 9.0 "Package Specifications"	• No changes.
Section Appendix A "Support Documentation"	• Added SC2200UFH-300 to Section A.1 "Order Information".

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 87 90

National Semiconductor Asia Pacific
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com

www.national.com

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.